COMMODORE USERS

48355
K

Ahoy!

$2.50/CAN.$3.00 APRIL 1985



0 71896 48355 04

"The Complete CARDCO Line"

. . . and still growing!

CARDCO provides "Commodore-ready" computer accessories that will enhance your utilization of Commodore-64 and VIC-20 Computers, increase their capability, and add to your enjoyment and skill. AND, they're available for use with other personal computers, too.

Designed with the user in mind, CARDCO offers fine accessories including Printer Interfaces with and without graphics, Expansion Interfaces, Memory Expansions, Cassette Interfaces, Numeric Keypads PLUS "NOW" Software for your VIC-20 and C-64. These programs include the "WRITE NOW" Word Processor, "MAIL NOW" Mailing List, PRINTER UTILITY PROGRAMS on Tape and on Disk, "SPELL NOW" Spell Checker, "GRAPH NOW" including "PAINT NOW", and "FILE NOW".

CARDCO has three new Letter Quality PRINTERS with your choice of drumhead design (8 1/2" carriage), Daisy Wheel Design (13 inch carriage) and Daisy Wheel Design (11 inch carriage). "Commodore-ready" . . . plus; with compatible input for PC, PC jr., TRS-80 and many more personal computers. CARDCO's NEW "DATA CASSETTE RECORDER/PLAYER" is also "Commodore-ready" and ready for instant shipment at prices that will amaze you.

CARDCO will constantly increase its line with unique and new products to enhance the enjoyment of computer owners.

Write for illustrated literature and prices or see CARDCO Computer Accessories and Software wherever Computers are sold.



cardco, inc.

300 S. Topeka Wichita, Kansas 67202 (316) 267-6525

"The world's largest manufacturer of Commodore accessories."

Commodore™ is a registered trademark of Commodore Business Systems, Inc.

Reader Service No. 103

Ahoy!

CONTENTS

DEPARTMENTS

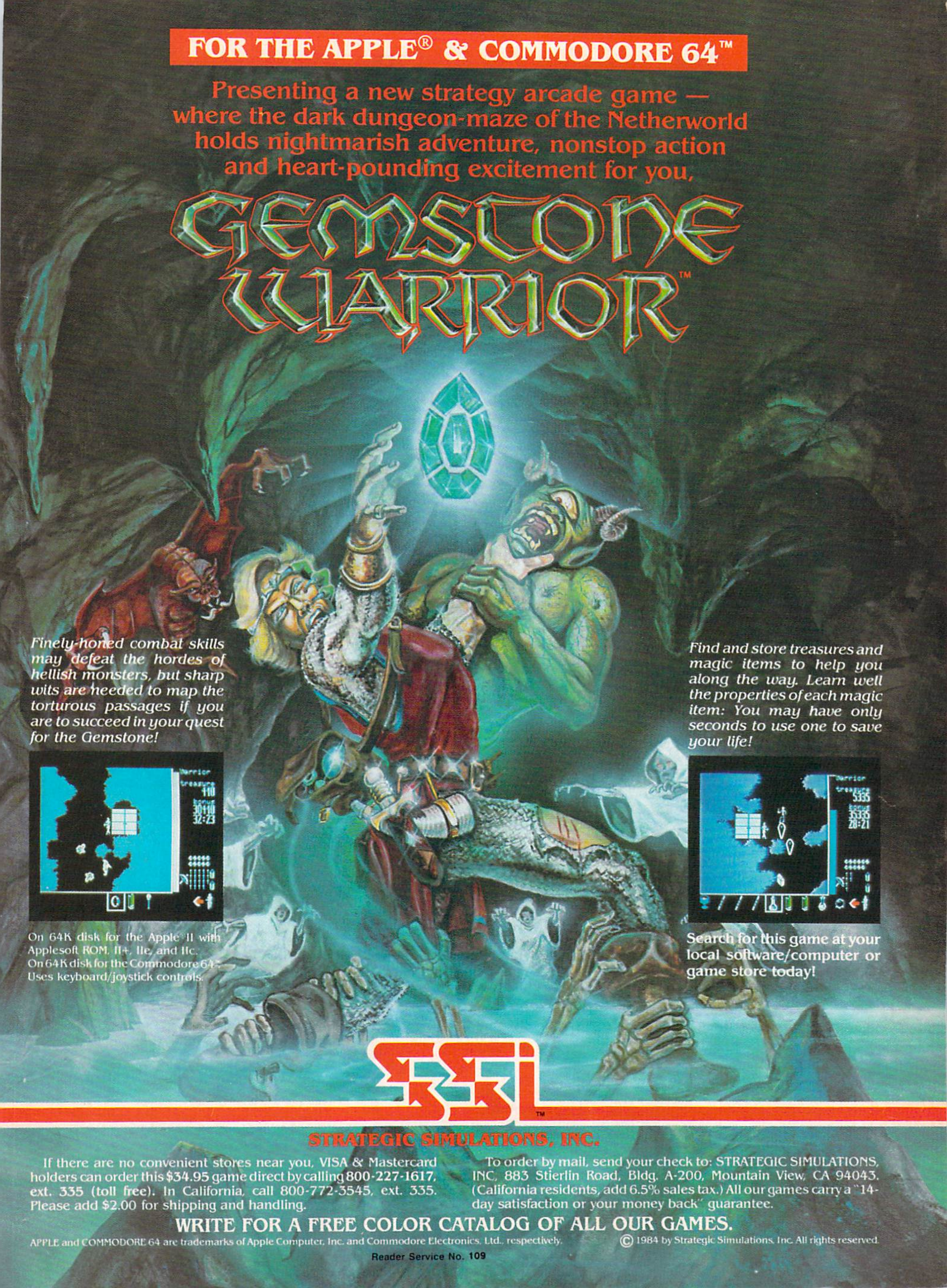
FEATURES

PROGRAMS

Cover illustration by James Regan

AHOY! No. 16, April 1985. Published monthly by Ion International Inc., 45 W. 34th St., Suite 407, New York, NY, 10001. Subscription rate: 12 issues for $19.95, 24 issues for $37.95 (Canada $26.95 and $49.95 respectively). Application to mail at 2nd Class postage rates is pending at New York, NY and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street, Suite 407, New York, NY 10001. Direct all address changes or matters concerning your subscriptions to *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054. All editorial inquiries and software and hardware to be reviewed should be sent to 45 W. 34th St., Suite 407, New York, NY 10001.


FOR THE APPLE® & COMMODORE 64™
Presenting a new strategy arcade game — where the dark dungeon-maze of the Netherworld holds nightmarish adventure, nonstop action and heart-pounding excitement for you.
GEMSTONE WARRIOR™
Finely-honed combat skills may defeat the hordes of hellish monsters, but sharp wits are needed to map the torturous passages if you are to succeed in your quest for the Gemstone!
On 64K disk for the Apple II with Applesoft ROM, II+, IIe, and IIc.
On 64K disk for the Commodore 64™.
Uses keyboard/joystick controls.
Find and store treasures and magic items to help you along the way. Learn well the properties of each magic item: You may have only seconds to use one to save your life!
Search for this game at your local software/computer or game store today!
SSI™
STRATEGIC SIMULATIONS, INC.
If there are no convenient stores near you, VISA & Mastercard holders can order this $34.95 game direct by calling 800-227-1617, ext. 335 (toll free). In California, call 800-772-3545, ext. 335. Please add $2.00 for shipping and handling.
To order by mail, send your check to: STRATEGIC SIMULATIONS, INC, 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, add 6.5% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.
WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.
APPLE and COMMODORE 64 are trademarks of Apple Computer, Inc. and Commodore Electronics, Ltd., respectively.
© 1984 by Strategic Simulations, Inc. All rights reserved.
Reader Service No. 109

VIEW FROM THE BRIDGE

Sometimes it pays to put things off. If you've put off learning machine/assembly language until now, you waited just long enough—because this issue marks the beginning of our headlong plunge back to programming's *true* basics!

But we didn't want just *anyone* teaching our readers machine language. So we went out and hired the best—Mark Andrews, author of *Commodore Roots: Assembly Language Programming for the Commodore 64* (Howard W. Sams & Co.) and one of the most respected writers in the field. His *Commodore Roots* series premiering in this issue is not an adaptation of his latest book, but an original column written exclusively for *Ahoy!* If you follow *Commodore Roots* each month, you could be a proficient assembly language programmer in far less time than it took you to catch onto BASIC. (Turn to page 51.)

For those of you who went ahead without us and began learning machine language on your own—we praise your initiative! And we reward you with the first of several *Rupert Reports* dealing with machine language on a slightly more advanced level. Dale's explanation of *Getting into the Kernal* should be understandable by beginners as well. (Turn to page 45.)

As further cogs in your ML education, we present reviews of the *Merlin 64* assembler (page 26) and *Assembly Language for Kids* (page 40)—plus the following article/programs:

• *BASIC Trace* by Daniel M. Green will follow the flow of your program as it executes, an invaluable aid for debugging and general education. (Turn to page 57.)

• How fast is machine language? One illustration is provided by Anthony Wood's *Faster 64,* which can speed up your BASIC programs by as much as 45%. (Turn to page 39.)

Among the other programs in this issue of *Ahoy!*:

• *Booter* by Daniel M. Green will write a fast ML boot that will automatically load and run any VIC or 64 program you specify. (Turn to page 41.)

• *Elecheck* by Glenn Lumpkins (author of *Insurance Agent*) provides an easy-to-use and versatile checkbook management system for the 64. (Turn to page 43.)

• *Roll Over Pachelbel* by Simon Edgeworth gives you the chance to experiment with 12 different verses of the famous composer's Canon. (Turn to page 91.)

• None of those user-friendly features of modern text adventures in Anthony Wood's *Space Hunt*—you'll rough it in the tradition of the classic adventure games! (Turn to page 43.)

• *Hop Around* by Kevin Dewey (author of *To The Top*) offers equal portions of exciting gameplay and outright silliness. For the VIC and 64. (Turn to page 86.)

With a lineup of programs this exciting, it's no wonder that we're halfway through this page without even mentioning the inclusion in this issue of the most requested feature article in our history! You've besieged us with requests for a comparison of the various quick-loading programs available for the 1541. We've gone one better, providing a two-part profile of not only the enhancements, but the *substitutes* available for the Achilles' Heel of the Commodore hardware line. This issue, Morton Kevelson (who else?) takes apart the Indus GT and Commander II disk drives and examines some other useful disk drive utilities in *Disk Spinners* (turn to page 29)... next month it's on to Fast Load, *Kwik-Load!*, and the 1541 Flash. He kicks off the whole affair by issuing hardware manufacturers *The Disk Drive Challenge* (turn to page 6).

Also from Morton comes an addendum to his printer interfacing review concluded last issue—a look at the just-completed Cardco OuiG. (Turn to page 88.)

Orson Scott Card continues tutoring you on *Creating Your Own Games on the VIC and 64* with an explanation of *The Joy of Sticks*—and the misery inherent in programming with them. You'll also enjoy Orson Scott's segue into his vision of the Perfect Computer. (Turn to page 18.)

Most of our regular columns can also be found between these covers, including *S.O.S.* (solutions to reader's problems—see page 50) and *Ship to Shore* (featuring Cheryl Peterson's evaluation of the importance of SYSOP associations—see page 60). Two columns, however, deserve special mention this month—*Scuttlebutt* for its analysis of the new 128PC and LCD Portable computers from Commodore, and its listing of the coming Commodore-compatible releases announced at January's Consumer Electronics Show; and *Reviews,* for the game-reviewing debut of Arnie Katz, founder and former editor of *Electronic Games* magazine and the industry's noted "guru of gaming."

We go to press amid disturbing news reports of plummeting earnings for Commodore International—a drop of 94% in earnings for the second quarter ended Dec. 31, according to the *Wall Street Journal.* That translates into a drop in net income from $50.1 to $3.2 million, and in sales from $431.4 to $338.7 million.

What does that translate into for the Commodore user? Nothing concrete enough to push the panic button over. The price of the C-64 will drop sharply, especially as Commodore attempts to liquidate its huge inventory in preparation for the 128PC. Software and peripherals manufacturers may hesitate before putting all their eggs in the Commodore basket, in view of the revelation that the C-64 user base will not increase until infinity.

Or, Commodore's next quarter could be their best ever. Keep computing and watch for further announcements!

—David Allikas

DISK DRIVE CHALLENGE

By Morton Kevelson

This month and next we report on some new products for the Commodore 64, all with one thing in common: they seek to improve upon the performance of the 1541 disk drive provided by Commodore. These products span all possibilities, starting with alternative drives, progressing to firmware modification of both the 1541 and the computer, and settling down with LOADable software that achieves a variety of effects. Most of these products present a common theme: a way to let the user spend less time waiting for the disk drive. As all users of the Commodore drive are well aware, it is not the fastest system on the market.

Indus Systems, with their Indus GT disk drive, introduce a new concept for the Commodore system—a disk drive emulator in ROM, in the disk drive itself. It is a good idea. Without question, having a set of frequently used utilities available in this fashion is extremely convenient.

However, it seemed to me that this concept should be taken just a bit further. The basic premise is that mechanical hardware is expensive and that silicon (or semiconductor chips) are cheap. How about setting up a RAM disk with the capacity of a Commodore formatted floppy disk, a full 170 kilobytes, in the disk drive itself as drive 1? Properly configured, this system would behave just like a dual disk drive. In some ways it could be expected to outperform a dual drive.

Take the matter of backing up a disk. Having an internal parallel link between RAM and floppy should allow for the high speed transfer of the entire contents of a disk in less than one minute. This is based on performance already exhibited by the MSD SD-2 dual disk drive. This would allow a backup copy to be made in well under two minutes. Furthermore, once the data is in RAM, additional copies can be made in just a matter of seconds. An arrangement of this sort would also be a practical way to set up a large program, with multiple disk loaded modules, while freeing up the mechanical drive for data storage. Conversely, the RAM system can be used as an online data buffer for use with a copy protected program disk. When the work session is over, the data can be transferred to a floppy disk. Finally, the inclusion of a battery backup system would permit the semi-permanent storage of a disk full of programs in the second drive, even after the rest of the system is shut down.

Of course this is all just speculation, but with the way things are going, who knows? ☐

SEE *DISK SPINNERS,* PAGE 29

CALLING ALL STARVING COMPUTER ARTISTS...

If you're in need of a square meal, we can't help you. But if it's publicity you're starved for, we've got just the ticket. In future issues, *Ahoy!* will feature a gallery of the finest computer graphics our readers can generate.

We invite you to send your best work on disk, accompanied by a stamped and self-addressed mailer, to Morton Kevelson, P.O. Box 260, Homecrest Station, Brooklyn, NY 11229. Indicate the drawing package that was used to create the image. If you employed a bit map of your own design, indicate the appropriate file parameters, i.e., hi-res or multicolor, location of bit map, screen and color data.

Send along your latest masterpiece today. Who *knows* who might be reading?



QUESTRON
Live the Fantasy and the Adventure.
APPLE,® ATARI® & COMMODORE 64™ versions now showing at a computer/ software or game store near you.
STRATEGIC SIMULATIONS INC. PRESENTS A FANTASY ADVENTURE GAME: QUESTRON™
ONE OF THE FINEST CHAPTERS IN THE NEVER-ENDING SAGA OF THE BATTLE BETWEEN GOOD AND EVIL
Starring YOU as THE HERO • MESRON, THE GOOD WIZARD • MANTOR, THE EVIL SORCEROR
AND HIS HORDES OF HERO-CRUNCHING MONSTERS • Written and directed by CHARLES DOUGHERTY
On 48K disk for your Apple® II with Applesoft ROM card, Apple II+, IIe, or Apple III. Also for Atari® home computers.
PG
THIS GAME RATED POSITIVELY GREAT.
Ideal for Fantasy Adventurers of all ages.
APPLE & ATARI are trademarks of Apple Computer, Inc. and Atari, Inc., respectively.
STRATEGIC SIMULATIONS INC
COMMODORE 64 is a trademark of Commodore Electronics, Ltd.
If there are no convenient stores near you, VISA & Mastercard holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, call 800-772-3545, ext. 335. QUESTRON™ goes for $49.95, plus $2.00 for shipping and handling. (C-64 version is $39.95.)
To order by mail, send your check to: STRATEGIC SIMULATIONS INC. 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043. (California residents, please add 6.5% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.
WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.
Reader Service No. 175

EXPAND YOUR

H O R I Z O N

ApSoft-64

The horizons have been expanded. Thousands of Applesoft programs have been added to the Commodore-64 software base. Priced at $39.95, ApSoft-64, the ultimate BASIC extension, makes your 64 think it's an Apple.

A variety of valuable programs, including "File Cabinet," the renowned Applesoft data base management program, are provided on the ApSoft-64 disk at no extra charge. Hundreds more are available directly from FSI Software at $1.00 each.

FSI
SOFTWARE

FSI Software
P.O. Box 7096,
Minneapolis, MN 55407

Dealer and distributor inquiries invited.

Toll-free, 1-800-328-0145,
In MN: (612) 871-4505

Apple is a registered trademark of Apple Computer Corp.

Commodore-64 is a registered trademark of Commodore Business Machines, Inc.

Reader Service No. 110

SCUTTLEBUTT

HARD DRIVE • PROMAL PACKAGE • TEN NEW PRINTERS • DIGITIZER • TEXT & GRAPHICS CARD • BULLETIN BOARD NEWS • COMPUTER DIET • EDUCATIONAL & GAME RELEASES • MUSIC PROGRAMS • PRINTER BUFFER • COMMODORE 128PC & LCD PORTABLE COMPUTERS, AND MORE CES NEWS

WINTER '85 CES—AN OVERVIEW

The Winter Consumer Electronics Show was, this past January as every January, the showcase for what's ahead for the home electronics industry in the coming year. Most of the Commodore-compatible products newly announced there appear in this edition (and last month's edition) of *Scuttlebutt*.

Just how many new 64-compatible releases are on their way to your dealer's shelves? More than your dealer's shelves can hold, particularly in the educational area. Most of this, alas, is just someone else's word processor, someone else's typing program—the innovative releases are, as always, in the minority.

If you're a VIC owner, don't waste the trip to your dealer in the months to come, unless it's to buy a new computer. Some new VIC releases will show up, but they'll be scarcer than 2-track tapes.

An overview of the January '85 CES must make mention of the excitement generated by Commodore—and Atari. We've been following the fueling battle between the two low-end giants in these pages, and looking forward to their CES faceoff. The fight fan in us was not disappointed.



Atari 65 XE: new C-64 rival.

Commodore stole the show at last January's CES by their very presence, striding into town like MacArthur into the Philippines after their total victory in the '83 Home Computer Wars—and premiering the Plus/4 (then the 264) to boot.

This year, both Commodore and Atari had impressive wares to hawk. First, the challenger: Atari announced six new computers—their new flagship 65 XE with 65,536K RAM, 11 graphic modes, 256 colors, 4 independent sound voices, and built-in BASIC; the 65 XEM with 8 independent voices; the 65 XEP, a portable with built-in 5" 40/20 column monochrome monitor with built-in 3½" disk drive; the 130 XE with 131,072K RAM; and the 130 ST and 520 ST personal computers, packed, respectively, with 131,072 and 524,288K RAM.

You can read all about Commodore's entries, the 128 Personal Computer and LCD Portable, below (and in Morton Kevelson's usual microscopic analysis in an upcoming issue).

Commodore trumpeted their new releases with billboards proclaiming "Bad news for IBM and Apple." Whether or not the 128PC can cut into sales of the thriving PCjr and IIc remains to be seen, but it's certain that Commodore cannot avoid renewed and costly competition with Atari.



The Commodore 128PC is actually three discrete computers in a single box. *READER SERVICE NO. 135*

COMMODORE ANNOUNCES FOUR NEW COMPUTERS AT CES

Now that we have your undivided attention, we'll explain. No, we don't know something that everyone else, including Commodore, does not. Three of these computers will actually be sold in a single box as the Commodore 128 Personal Computer. The fourth is the Commodore LCD lap portable. Both of these machines are best described as a consolidation of existing technology rather than a presentation of new and innovative designs. By so consolidating, Commodore is able to offer the user more features at a lower cost than ever be-



The Commodore LCD Portable's screen has a legible 80-column display. *READER SERVICE NO. 136*

fore. What is even more important is that full compatibility will be maintained with two of the largest existing software bases, that of the Commodore 64 and the CP/M operating system.

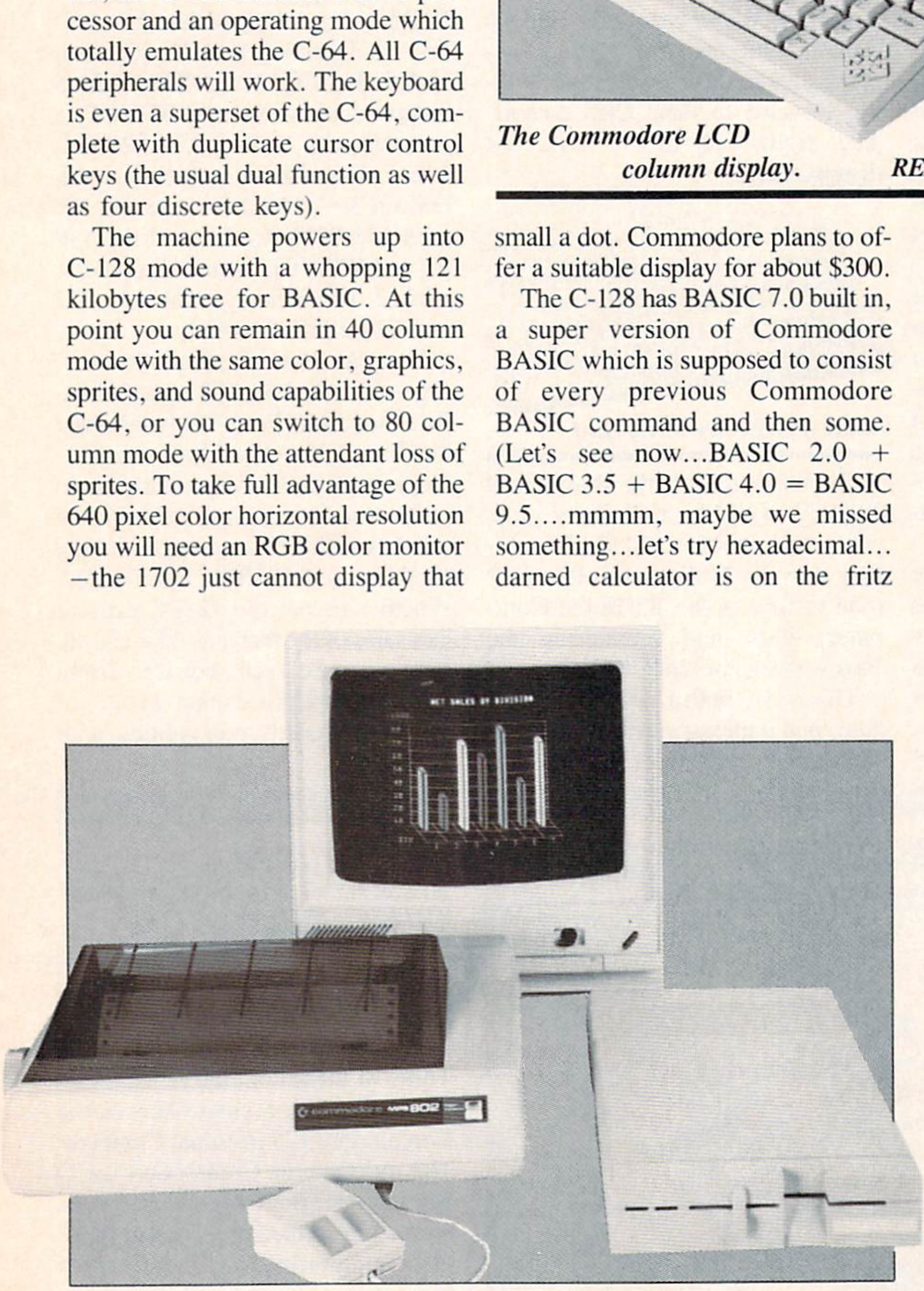
The C-128PC is actually three discrete computers in a single case sharing a common keyboard. To maintain upward compatibility with the more than three and a half million Commodore 64's already on the market, the C-128 includes a 6510 processor and an operating mode which totally emulates the C-64. All C-64 peripherals will work. The keyboard is even a superset of the C-64, complete with duplicate cursor control keys (the usual dual function as well as four discrete keys).

The machine powers up into C-128 mode with a whopping 121 kilobytes free for BASIC. At this point you can remain in 40 column mode with the same color, graphics, sprites, and sound capabilities of the C-64, or you can switch to 80 column mode with the attendant loss of sprites. To take full advantage of the 640 pixel color horizontal resolution you will need an RGB color monitor —the 1702 just cannot display that small a dot. Commodore plans to offer a suitable display for about $300.

The C-128 has BASIC 7.0 built in, a super version of Commodore BASIC which is supposed to consist of every previous Commodore BASIC command and then some. (Let's see now...BASIC 2.0 + BASIC 3.5 + BASIC 4.0 = BASIC 9.5....mmmm, maybe we missed something...let's try hexadecimal... darned calculator is on the fritz again...some new Commodore special math...) An 8502 microprocessor, an HMOS version of the eight bit 6502, provides the brains behind the operation. Some sophisticated bank switching is incorporated into the design to manage the RAM and 48 kilobytes of ROM plus 16 kilobytes of DOS enhancement ROM which make up the computer's operating system. The C-128 will be expandable to 512 kilobytes via an optional RAM disk.

The C-128 can be switched into Commodore 64 mode, whereby for all practical purposes it becomes a Commodore 64. Even the 14-key numeric keypad is disabled, since these keys return a different CHR$ code than the original number keys. The keypad can be reenabled via a soft switch; however, existing software will not recognize it without some modification. The computer reverts back to BASIC 2.0 running on 16 kilobytes of ROM and 64 kilobytes of RAM in only the 40 colum mode. Control is turned over to a built-in 6510 microprocessor. The entire machine will have to be turned off and on to leave C-64 mode.

The third built-in machine runs CP/M Plus version 3.0 on a 4 MHz Z80A microprocessor. The new 1571 disk drive (see below) will be required to allow the machine to read IBM System 34 format CP/M disks, the same format used by the Kaypro



New peripherals: printer, RGB color monitor, half-height disk drive.
READER SERVICE NO. 137

and Osborne computers. In this mode both 40 and 80 column operation will be available, along with 128 kilobytes of RAM expandable to 512 kilobytes on RAM disk.

THE LCD PORTABLE

This five-pound five-ounce machine features a very readable 80 column by 16 line display (480 by 128 pixels in graphics mode). Although this is not a full-screen 25-line display, it still allows for 28 percent more information than what we are used to with the C-64. Built in software on 96 kilobytes of ROM will include BASIC 3.6 (another number), word processor, spreadsheet, file manager, address book, scheduler, calculator, memo pad, terminal emulator, and a machine language monitor. Only 32 kilobytes of RAM are included, but this may be increased in the final version. The brains behind this operation is a 65C102 low power CMOS eight bit microprocessor.

The machine will include a built-in 300 baud modem and an A/C power adapter. The 500 milliwatt power requirements should permit up to 15 hours of stand alone operation on four AA alkaline batteries. The computer was displayed at CES with a new 3.5 inch microfloppy which will also be battery powered. Compatibility will be maintained with all existing Commodore serial bus peripherals.

THE 1571 DISK DRIVE

The key to the power and performance of the C-128 rests in this new disk drive. Like the C-128, the 1571 has three operating modes. In either of the two Commodore modes (C-64 or C-128) it will operate as a 350 kilobyte double sided single density drive. Each single side of the formatted disk will be fully compatible with the existing 1541 format.

For the C-64 operating mode, data will be transferred at the existing 300 character per second (cps) rate. In the C-128 operating mode the transfer rate goes to 1500 cps, a five to one improvement.

In CP/M mode, operation speeds up to 3500 cps with a burst rate up to 4000 cps possible, a better than thirteen time improvement over the original 1541 mode. Data storage also goes to 410 kilobytes double sided double density in CP/M mode.

The internal electronics are similar to the 1541's, with two kilobytes of RAM and a 6502 microprocessor. The built-in disk operating system has been expanded to 32 kilobytes (the 1541 has only 16) on ROM in support of the increased storage capacity, higher speed, and the three operating modes. The computer connection is via the same type of serial port used by the 1541. Note that the high speed modes will be available only when used with the C-128.

IN CONFUSION

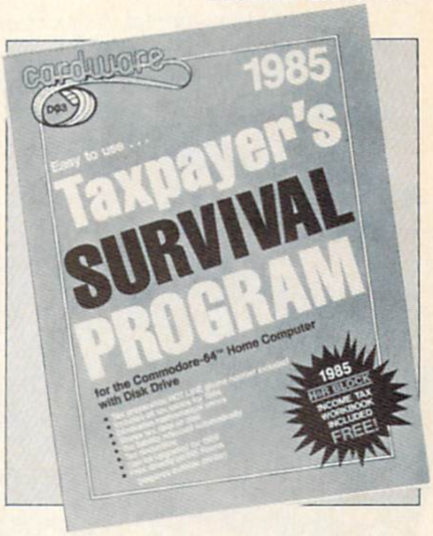
At a projected selling price of under $350, the C-128 is an excellent way to upgrade to a more powerful system without sacrificing a single dime of existing investment in hardware and software. Utilizing the C-128 to its fullest extent will be a bit more costly. At present the projected system price for the computer, 1571 disk drive, and RGBI monitor approaches $1000. Of course, 80 column monochrome capability can be had for an additional investment of under $100.

At under $550, the LCD portable is quite reasonable—at today's prices. It certain does not lack for features and convenience.

The big question in our minds is, where does all this leave the C-64 and the fledgling Plus/4 and C-16 computers? Given the capabilities of the C-128, will the C-64 be this year's $100 computer? Under this scenario there hardly seems to be a place for the Plus/4 in Commodore's scheme of things. Perhaps a harbinger of things to come is the less than $560 price advertised by one New York retailer for a C-64 with 1541 and 1702 as of this writing (January).

The ultimate success (and the actual selling price) of a C-128 system may well be beyond Commodore's control. If Jack Tramiel, Commodore's former chief exec, makes good on his Atari intentions, we may well be treated to another episode in the ongoing Computer Wars saga.

—Morton Kevelson



READER SERVICE NO. 138

TAX SURVIVAL

Cardco's *1985 Taxpayer's Survival Program* automatically adjusts tax liability, eliminates mathematical errors, and stores tax data on disk. Featuring tax tables for 1984, the program can be updated for next year as well. Price: $39.95.

Cardco, Inc., 300 S. Topeka, Wichita, KS 67202 (phone: 316-267-3807).

SUPERCARD

The Super 80 text and graphics card, featuring built-in word processing and spreadsheet programs, generates 80-column text which can be superimposed over color graphics. Scheduled for March availability, at a price of $99.95.

Personal Peripherals Inc., 1505 S. Green, Longview, TX 75602 (phone: 214-758-8874).

HIGH-SPEED LANGUAGE

Systems Management Associates claim that the new PROMAL language (similar to C and PASCAL) is 70-2000% faster than BASIC, COMAL, FORTH, and PASCAL. Their PROMAL package provides C-64 users with a one-pass compiler, a full screen editor, a command executive (operating system), and a library of predefined utility subrou-

tines. Retail price is $49.95, or $99.95 for Developer's Version (including unlimited run-time distribution license).

Systems Management Associates, 3700 Computer Drive, P.O. Box 20025, Raleigh, NC 27619 (phone: 919-787-7703).



Has expandable 700-food database.
READER SERVICE NO. 129

MASS REDUCTION

Developed by a team of clinicians and scientists, the *Original Boston Computer Diet* provides a personalized approach to weight loss and fitness. The dieter starts by providing a complete medical history and review of eating habits; subsequent sessions with a simulated weight loss counselor provide food intake analysis and menu planning. A database of 700 foods can be expanded to accommodate personal preferences. For the C-64; $49.95.

Scarborough Systems, Inc., 25 N. Broadway, Tarrytown, NY 10591 (phone: 914-332-4545).

DIGITIZER FOR C-64

The Computereyes video acquisition system plugs into the C-64's user port and connects to any standard video source (VCR, video camera, etc.), allowing the user to capture real-world images on the 64's high-resolution graphics display. A black and white image can be acquired in under six seconds; multiscan mode can provide grey-scale images.



Computereyes interface module, software, printer dump. READER SERVICE NO. 130

Price of the Computereyes interface module and software is $129.95 plus $4.00 shipping; with video camera, $349.95 plus $9.00 shipping (MA residents add 5%) from Digital Vision, Inc., 14 Oak Street—Suite 2, Needham, MA 02192 (phone: 617-444-9040).



Create coloring books and pictures.
READER SERVICE NO. 131

IT'S EDUCATIONAL

We nearly omitted this item out of spite, remembering all the high school newsletters we had to type on mimeograph stencils. The present generation of student (or club or community) journalists will have it easier thanks to *The Newsroom,* soon to be released for the C-64. The user may type stories in one of five fonts with the program's built-in word processor, choose from a variety of headlines, borders, and tones, and insert clip art from a file of over 600 pieces. Type can wrap automatically around the art.

Springboard Software, Inc., 7807 Creekridge Circle, Minneapolis, MN 55435 (phone: 612-944-3912).

Three additions to Mindscape's Sprout line for children aged four to eight: *Castle Clobber* (help Tonk rescue all the toys in Tink!Tonk! land from the evil Gork), *Subtraction Fair* (five game booths featuring subtraction contests), and *Color Me: The Computer Coloring Kit* (design and print out pictures, coloring books, and stickers). First two $24.95, *Color Me* $29.95; all for the 64.

Mindscape Inc., 3444 Dundee Road, Northbrook, IL 60062 (phone: 312-480-7667).

Mimi lets children aged two and up create stories by linking together the music and graphics sequences that appear on the computer screen at the touch of a key. On C-64 disk, in English or French: $29.95.

Logidisque Inc., C.P. 485, succursale Place d'Armes, Montreal, Quebec, Canada H2Y 3H3 (phone: 514-842-5221).

Three for the 64 from DesignWare: *The Body Transparent* ($44.95) requires students aged 9 to 16 to move organs and bones into their correct position on the male or female body. *European Nations & Locations* (ages 9 to adult; $44.95) teaches geographic and historic facts in a trivia game format.

Remember! ($79.95) provides long-term retention techniques for students of history, chemistry, foreign languages, and a variety of other subjects.

DesignWare, 185 Berry Street, San Francisco, CA 94107 (phone: 415-546-1866).

The *Gruneberg Linkword Language Course* promises to teach students of Spanish, German, French, or Italian a vocabulary of about 400 words and a basic grammar in as little as 10 hours. Each package contains program (on disk or cassette for the 64, cassettes only for the VIC 20 +16K) plus audio cassette.

Audiogenic Ltd., 39, Suttons Industrial Park, London Road, Reading, Berkshire, England RG6 1AZ (phone: Reading (0734) 664646).

Two for the 64 from Simon & Schuster's Electronic Publishing Group:

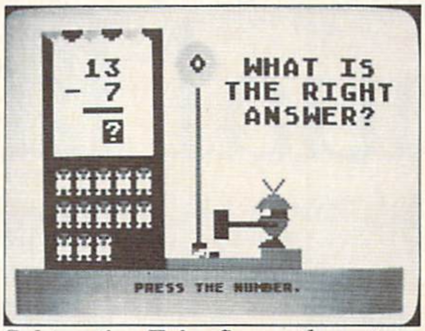
Chem Lab poses fifty problems that require children aged 9 to 13 to mix, heat, and combine chemicals onscreen, using two robot arms and assorted laboratory equipment. Price: $39.95.

Lovejoy's Preparation for the SAT includes tutorials, practice tests, and test-taking tips and techniques, along with the concise version of Lovejoy's College Guide. Price: $8.95.

Simon & Schuster Electronic Publishing Group, 1230 Avenue of the Americas, New York, NY 10020 (phone: 212-245-6400).

PASCAL: A Modern Programming Language, a set of five filmstrips and five audio cassettes designed for students with or without BASIC knowledge, can be obtained for $125.00 from Educational Activities, Inc., P.O. Box 392, Freeport, NY 11520.

Bible-Grams challenges two to ten participants to identify Biblical anagrams. For the 64 or VIC 20 (+16K) on cassette ($24.95) or disk ($29.95). Additional anagram searches are available on *Bible-Grams2* and *3*, each selling for $12.95 (cassette) or $17.95 (disk).



Subtraction Fair: ***five math contests.***
READER SERVICE NO. 132

SEI Enterprises Inc., 17 Serpi Road, Highland Mills, NY 10930 (phone: 516-499-2525).

New and coming releases for the C-64 from Grolier Electronic Publishing:

Step One ($39.95) provides an introduction to word processing and programming.

Friendly Filer ($39.95) acquaints students from the third grade up with database management. *Easy Graph* ($39.95) does likewise with computer graphing.

Friendly Files ($14.95) provides facts on science and nature and US and world information. (Available in April.)

Educalc ($49.95) introduces students from the eighth grade up to spreadsheet use. (Available in June.)

Grolier Electronic Publishing, Inc., 95 Madison Avenue, New York, NY 10016 (phone: 212-696-9750).

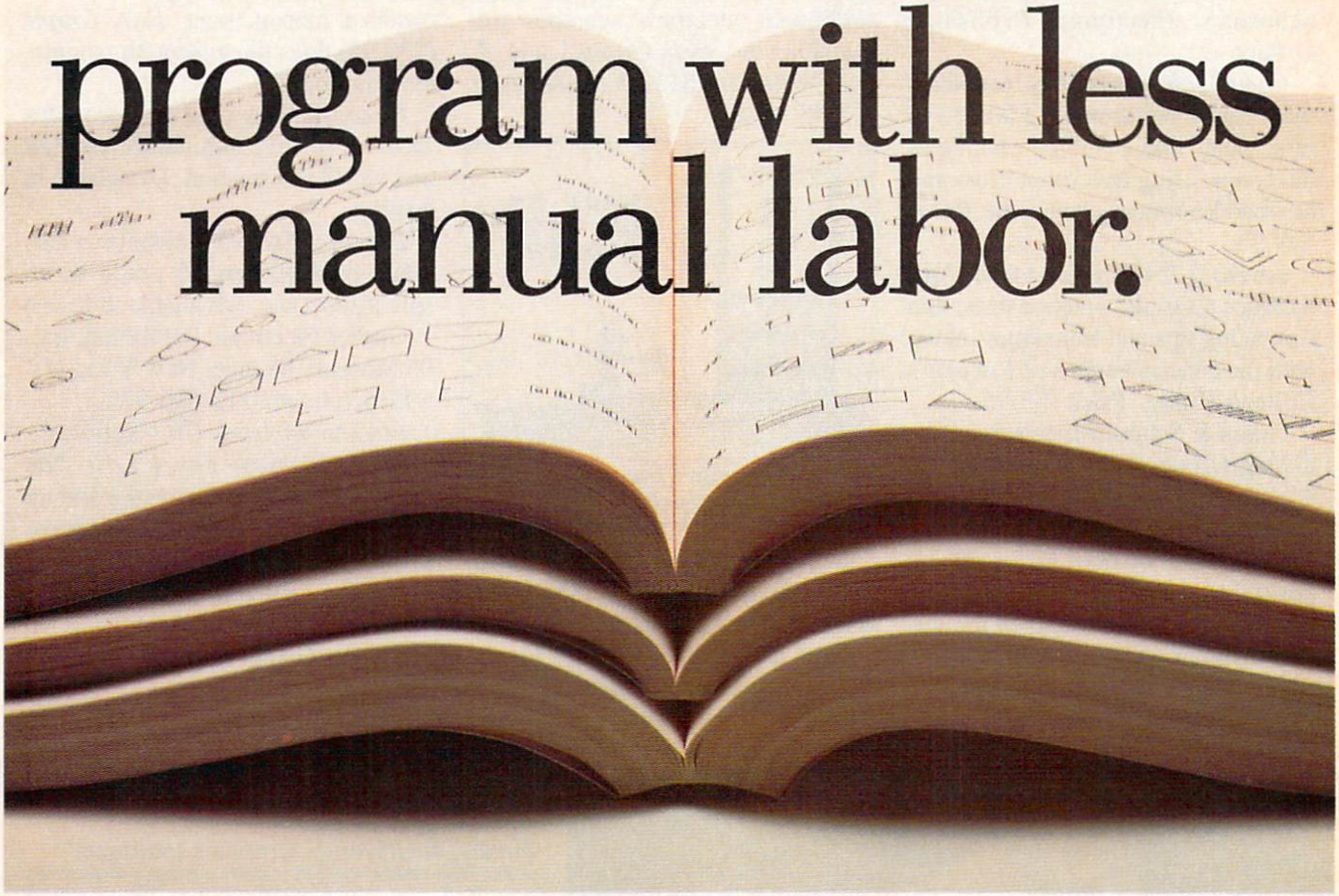
For the 64 from CBS Software:

Many Ways to Say I Love You ($29.95) lets children aged 4 and up create electronic greeting cards complete with animated graphics.

The Sea Voyagers ($39.95) chronicles the lives and discoveries of 30 New World explorers for ages 8 to adult.

Mastering the ACT ($150.00) drills

Choose the SAT program with less manual labor.



The idea behind choosing a computerized SAT program over a manual is to save you from piles of paperwork. But surprisingly, two of the best-known programs come with big, fat manuals and only 2 or 3 double-sided disks.

When that much information is put into the manual, what's left to put into the computer?

Why not buy a computer program that's really a computer program? Buy The Perfect Score from Mindscape for just $69.95.*



It has 6 double-sided disks and a real skinny manual. It even has print-out capability and a continuous on-screen clock. All this makes The Perfect Score more computerized than those others.

Now, if this cold logic fails to convince you, perhaps an emotional appeal to your sense of patriotism and social consciousness will. Your choice is this. Either you buy their SAT, which kills innocent trees to make all that paper. Or you buy our SAT with 6 disks and Save America's Trees.

The Perfect Score, $69.95, from Mindscape

MINDSCAPE INC

*Suggested retail price. Available for Apple,® Commodore 64™ and IBM.® The Perfect Score: Computer Preparation for the SAT. © 1984 Mindscape, Inc. All rights reserved.

Reader Service No. 146

high school students in the areas of English, math, social studies, and natural sciences. (Available in May.)

The *Success with Algebra* series, for grades 7 to 12, consists of four self-paced tutorials ($34.95 each): First Degree and Advanced Linear Equations, Binomial Multiplication and Factoring, Simultaneous Equations/Quadratic Equations, and Graphing Linear Equations.

Wordfinder ($34.95) requires players aged 10 and up to form as many 3-letter words as possible from one of 24 Key Words.

Mystery Master: Felony! ($34.95) puts playes aged 10 and up on the scent of 12 unsolved crimes, which they solve through interviewing suspects and searching locations (all the while improving their reading comprehension).

Interplanetary Pilot ($39.95) introduces the earthbound to aerospace science and navigation by putting them in control of a spaceship.

Quink (QUick thINKing) has players race to identify the member of a group of objects which does not belong. Price is $34.95.

The Railroad Works ($34.95) lets robber barons aged 10 and up build railroad empires across 12 screens.

The *Dr. Seuss Fix-Up the Mix-Up Puzzler* ($29.95) is an electronic jigsaw puzzle for ages 4 to 10, starring such characters as the Cat in the Hat.

CBS Software, One Fawcett Place, Greenwich, CT 06836 (phone: 203-622-2500).

NEW GAME RELEASES

The following new games are on disk for the C-64 only, unless otherwise noted:

You must survive such *Trolls and Tribulations* as deadly cretins and hungry buzzards as you comb eerie underground caverns in search of long-hidden treasure. Price: $24.95.

Creative Software, 230 East Caribbean Drive, Sunnyvale, CA 94089 (phone: 408-745-1655).

Not that the dedicated Commodore computerists among our readership have any, but *Idle Time* provides eight mentally challenging programs like Hangman (vocabulary), Number and Gunner (reasoning), and Secret Code (logic). Price is $19.95 plus $2.00 shipping/handling.

Able Software, P.O. Box 422, Kulpsville, PA 19443 (phone: 215-368-2518).

You're the *Suspect* in Infocom's latest text adventure, trying to clear yourself of a masquerade ball murder while finding out whodunit for your newspaper editor. The Advanced Level game retails for $39.95.

Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138 (phone: 617-492-1031).

Two games from Cosmi:

Richard Petty's Talladega pits the player against Petty and eighteen other top pro drivers in a 3D simulation of an actual NASCAR super speedway race. Price of cassette and disk "Double-Pak" is $19.95.

Super Huey teaches the player the techniques of rotary wing aviation, then sends him on four missions: solo flight, exploration, rescue, and combat. Price is $19.95.

Cosmi, 415 North Figueroa St., Wilmington, CA 90744 (phone: 213-835-9687).

Artworx has grouped five titles into their Gameline, available for $19.95 each: *Battle Through Time, Time Pilot* (both action/skill games involving offensive and defensive



The Artworx Gameline, packed in vinyl cases and priced at $19.95 each.
READER SERVICE NO. 141



Suspect ***requires you to solve—and clear yourself of—a society murder.***
READER SERVICE NO. 142

combat), *Alice in Videoland* (interactive graphic adventure based on Lewis Carroll's novel), *Ghost Chaser* (clobber phantoms with ectoplasm and search for keys to secret passageways in old mansion), and *Slap Shot Hockey* (two-player simulation with digitized speech).

Artworx Software Co., 150 North Main St., Fairport, NY 14450 (phone: 800-828-6573 or 716-425-2833).

Suncom has published a Bible edition for *PQ—The Party Quiz Game,* joining the supplemental disks covering education, entertainment, sports, and general knowledge, all retailing for $24.95.

Suncom, 260 Holbrook Drive, Wheeling, IL 60090 (phone: 312-459-8000).

British gamemaker Mastertronic has revised its list of initial US releases (reported in January's *Scuttlebutt*) to consist of *Chiller, 1985—The Day After, Kick Start, Challenger, Magic Carpet, BMX Racer, Dark Star, More Adventures of Big Mac, The Mad Maintenance Man, Mind Control,* and *Monty Python's The Quest for the Holy Grail.* Planned price has skyrocketed from $8.99 to $9.99 each (except *Holy Grail*—$12.00). Subsequent releases will include *The Wrath of Magra* and *The Games Creator,* selling for $19.99, and four additional programs at $12.99 each. Eight cassette-based VIC games will also be available, at $7.99 each.

Mastertronic, Inc., 6649 Odessa Avenue, Van Nuys, CA 91406 (phone: 818-780-9230).

Raiding the American market from even more distant shores in Radarsoft, Netherlands-based producer of *Maps 64 USA.* The cassette-based program requires you to pilot a helicopter across the country, identifying cities, rivers, and other landmarks.

US rep: Fischer Associates International, 4966 El Camino, Los Altos, CA 94022 (phone: 415-962-8216).

Epyx, Inc. has signed an agreement with Lucasfilm Ltd. to produce four games during the coming year. The first two will be *Ball Blazer* (futuristic fantasy sport played at very high speeds on a split screen) and *Rescue on Fractalus!* (space action-strategy game with 3D flight simulation.

Epyx, Inc., 1043 Kiel Court, Sunnyvale, CA 94089 (phone: 408-745-0700).

PRINTER INTERFACE

The Apricorn Parallel Printer Interface links a C-64, C-128, C-16, Plus/4, VIC 20, or SX-64 to a graphics printer, allowing it to emulate the Commodore character set. The unit consists of a Commodore Serial Bus connector, 6' cable, and Centronics type printer connector. Price: $69.95.

Apricorn, 7050 Convoy Court, San Diego, CA 92111 (phone: 619-569-9483).

C-64 HARD DRIVE

The ST10C 10 megabyte hard drive from Computer Specialties, Inc., compatible with all Commodore computers, can use either the serial or IEEE data transfer bus and features built-in backup mode, reformat protect, external device selection, and many other commands. Price will be in the $1500 range.

Also from CSI comes the IMP (Instructor/Monitor/Prompter) switching system that allows a teacher to link as many as 16 C-64's together and monitor them all from his desk, and the CSI Connect IEEE-488 interface that links the 64 to any Commodore IEEE-488 device.

Computer Specialties, Inc., P.O. Box 1718, Melbourne, FL 32902-1718 (phone: 305-725-6574).

PRINTER ENHANCER/ BUFFER

The Sprint Print printer enhancer and full-spooling buffer for the C-64 can increase the printing speed of the Commodore 1525 or 801 or various other printers by as much as 50%, in addition to providing a new character set with full descending characters and underlining capabilities. Buffer is available in 8, 16, or 24K. Price: $39.95.

Q R&D, One West Lake Street, Suite 320, Minneapolis, MN 55408 (phone: 612-922-7628).

WINGED WORDS

The *Fleet System 2* word processor for the 64 includes a built-in 70,000 word spell checker with capacity for 15,000 more words of the user's choice, plus 80-column capability, horizontal scrolling to 120 columns, and built-in mail merge. Price is $79.95.

Professional Software Inc., 51 Fremont Street, Needham, MA 02194 (phone: 617-444-5224).

WHAT'S IN A NAME?

The Handwriting Analyst ($99.95) includes a tutorial, reference materials, examples, templates, and C-64 software for enabling the user to determine personality characteristics from any handwriting samples. If that's not scientific enough for you, a color preference psychological test is also provided.

Franklin Software, P.O. Box 337, Blue Bell, PA 19422.

MUSICAL NOTES

The *Allegro* music and sound synthesis program lets beginners create music and incorporate it into their programs. Included are over 50 preprogrammed instruments and a notation language that permits rapid note entry and full-screen editing. The Commodore keyboard can also be used to play notes and chords in real time or accompany your music track. Price is $39.95.

Artworx Software Company, Inc., 150 North Main Street, Fairport, NY 14450 (phone: 800-828-6573 or 716-425-2833).

Broderbund will send a free $6.95 Hohner harmonica to anyone who purchases their *Music Shop* composer and synthesizer program and mails in the included coupon plus $1.50 for shipping.

Broderbund Software, 17 Paul Drive, San Rafael, CA 94903-2101 (phone: 415-479-1170).

Continued on page 84

PlayNET
announces 19 exciting ways to bring people together.

Now there's a Home Computer Network that lets you communicate with all kinds of people—all over the country! Make new friends, play exciting games, barter—shop—trade, all from the comfort of your home.

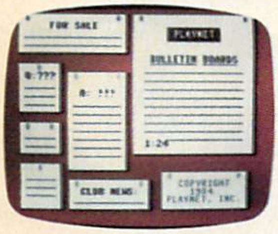
The network operates 6PM–7AM every weeknight, and 24 hours on Saturday, Sunday and Holidays. All you need to access PlayNET™ is a COMMODORE 64,* DISK DRIVE and MODEM.

1 TALK.
By typing on-line you can *talk to anyone,* or everyone, on the system. Meet fascinating people and make new friends from coast to coast.

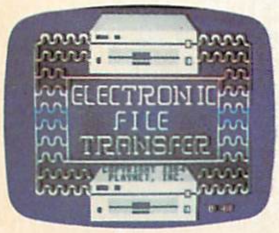


2 ELECTRONIC MAIL.
You can *send private messages* to people on the system, and the message will be waiting when they sign on!

3 BULLETIN BOARDS.
You can *post announcements,* or check the listings of other members. There are lots of boards for hobbies and special interests! If you don't find the one you're looking for—create your own!



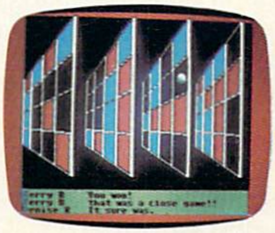
4 FILE TRANSFER.
You can even *transfer non-commercial programs* to other members! There is a small extra fee for this service.



5-17 GAMES! GAMES! GAMES!
PlayNET lets you play exciting games with real people, not just a computer. All our games have *full color* graphics, and they're all *interactive!*

- Plus the added feature of being able to talk with your opponent while you play. Discuss strategy, comment on moves, even try and psych your opponent out!
- New games are added all the time, and there are tournaments for every skill level.

CURRENT GAMES INCLUDE:





- Backgammon
- Boxes
- Capture the Flag
- Checkers
- Chess
- Chinese Checkers
- Contract Bridge
- Four-In-A-Row
- Go
- Hangman
- Quad 64™
- Reversi
- Sea Strike™

18 MORE! MORE! MORE!
There's *on-line updating* of your PlayNET Software as games and services are added. You have access to PlayNET's *Shopping Center* and *Information Center,* and every month you'll get our *Newsletter.*

19 YOUR SATISFACTION IS GUARANTEED for 30 days (or your full subscription price will be refunded upon receipt of the package).

This is all you pay:

- $39.95 for the PlayNET Software Package (3 disks and a User Manual), Monthly Newsletter, and 90 MINUTES ON-LINE TIME FREE!
- $6 monthly service charge.
- $2 an hour on-line time (the $2 includes the telecommunication charges). That's much less than a long distance phone call.



Here's how you can become a member!
It's easy to join, simply call PlayNET on the toll-free number, **1-800-PLAYNET,** or send the coupon below and access all your Commodore 64 has to offer. The sooner you do, the sooner PlayNET can put the whole country at your fingertips!

PlayNET
The Network that has people talking

48

SEND TO PLAYNET, INC.
P.O. BOX 587
WYNANTSKILL, N.Y. 12198

OR CALL 1-800-PLAYNET

YES! I WANT PlayNET TO PUT THE WHOLE COUNTRY AT MY FINGERTIPS. I UNDERSTAND THAT MY SATISFACTION IS GUARANTEED FOR 30 DAYS (or my full subscription price will be refunded upon return of the package) I may cancel my membership at any time by writing PlayNET.

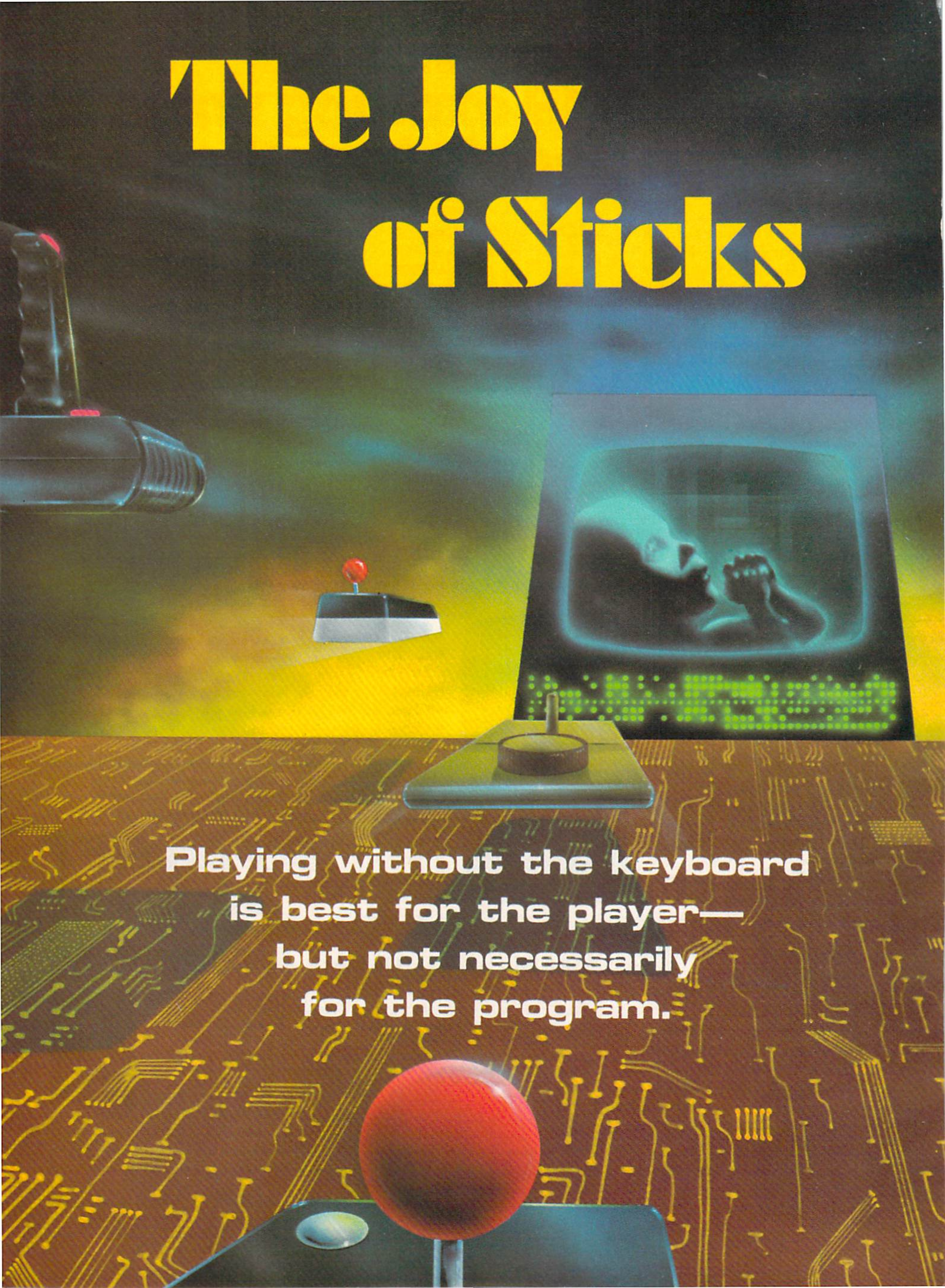
Bill me on my charge card for $39.95. (Plus sales tax for New York State residents). No checks, cash or money orders accepted. Please send me the PlayNET Software, User Manual, and 90 minutes of free on-line time.



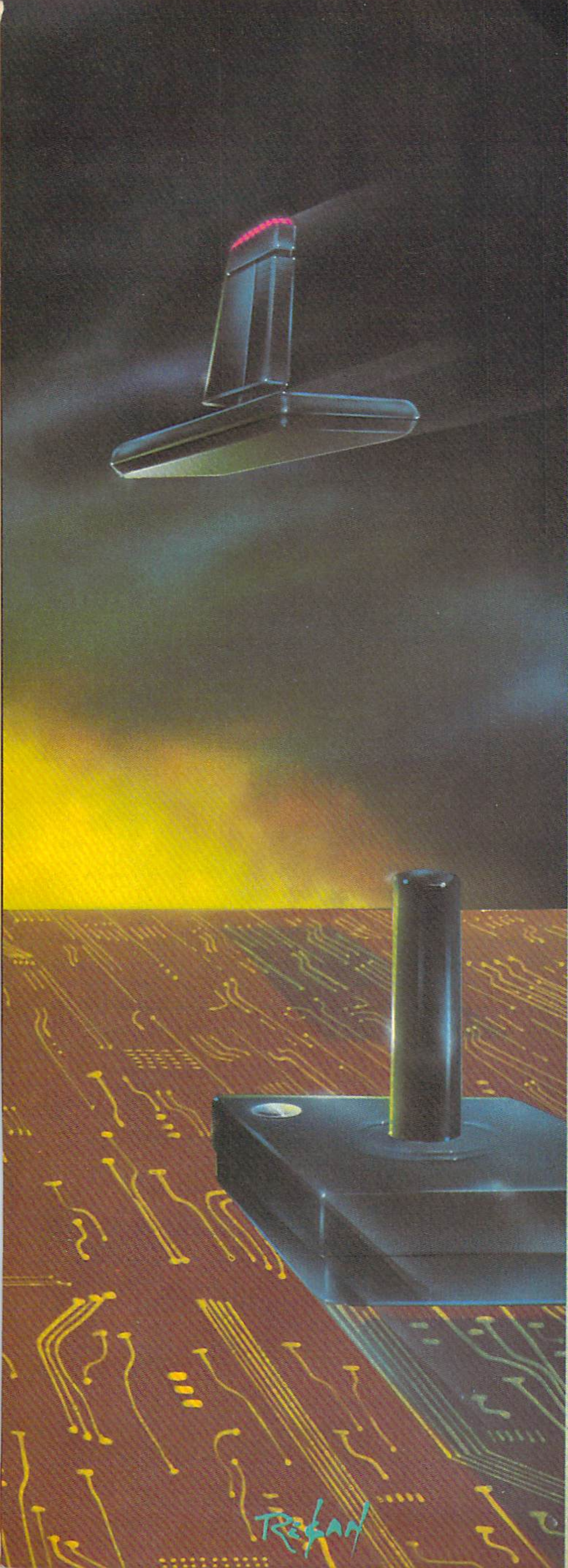
Please print.
Name__________
Address__________
City__________ State__________ Zip__________
Phone (   )__________
Check one: ☐ MasterCard ☐ VISA
Card #__________ Exp. Date__________
Signature__________

Reader Service No. 111

The Joy of Sticks



Playing without the keyboard is best for the player— but not necessarily for the program.



© James Regan 1984

CREATING YOUR OWN GAMES ON THE VIC AND 64

Ever since Atari first started selling little black boxes that played games, the joystick has been part of the ritual of computer games. Atari soon met deadly competition – but its joystick didn't! Commodore and Coleco both bowed to the supremacy of the Atari joystick and made sure their machines could use the same controller.

The joystick is the most logical device for manipulating a figure on the screen. Push the stick left, the figure moves left; push it to the right, and the figure moves right. Airplane pilots may be confused, however, since *they* pull on the joystick to go up, while gameplayers *push* on it to move the figure upward on the screen. Still, it's easy to understand.

When you let it go, the joystick centers itself and waits for the next command. It also has a button you can press. Also, by pushing the joystick in a direction halfway between two adjacent positions, you can send two signals at once – up and left, for instance – so that the program can receive and obey a diagonal movement instruction.

It's easy to understand, easy to use, and while it only allows five separate instructions, they are pretty much all you need in programming action games.

BITS CARRY THE MESSAGE

While Atari's BASIC included STICK and STRIG instructions to read the joystick and button, Commodore left us BASIC programmers on our own. (I'll talk more about decisions like that later on). So we have to use a PEEK statement in our programs to find out what message the player is sending with the joystick.

With the Commodore 64, this is pretty straightforward. Joystick 1, the forward joystick port, is read at location 56321. Joystick 2, plugged into the port right next to the on-off switch, is read at location 56320. To read what the joystick is "saying," you just use:

```
S1=PEEK(56321):S2=PEEK(56320)
```

The actual number means nothing. Instead, each *bit* of the number you get from the joystick port carries a separate message. You'll remember that a particular memory location has a maximum value of 255. This is because each location consists of eight bits, each of which holds either a 1 or a 0. This is interpreted as a binary number, like this:

bit:	7	6	5	4	3	2	1	0
value:	128	64	32	16	8	4	2	1

If a particular bit is *on*, or contains a 1, then the value for that bit is added to the overall total to get the decimal number. For instance, if bit 4 is on, you add 16 to the number. If bit 0 is on, you add 1. If they're all on, the total is 255. That's why 255 is the largest number you can store in a single byte.

by Orson Scott Card

BEFORE LOADING YOUR SOFTWARE
MAKE A PERFECT BACKUP WITH



THE AUTOMATIC DISK BACKUP
SYSTEM FOR YOUR 64 & 1541 DRIVE!

- Actually copies a FULL disk in only 4 MINUTES!
- Easiest to use — TOTALLY automatic copy utility
- State of the "ARTS" copy capability!
- UNLOCKS protected disks for backup copying
- Breaks through LATEST protection schemes
- Includes FAST copier, "ARTS" copier and TRUE Nibbler: you get 3 copiers for the price of 1
- HALF-TRACKING up to & including track 40!
- 100% machine code for fast, RELIABLE operation
- 4 min. copier AUTOMATICALLY puts "errors" on copy
- Fastest & most advanced copier you can buy
- Writes errors 20,21,22,23,27,& 29 WHILE COPYING!
- Transforms your 1541 into a SUPER-FAST copy drive!
- NO analysis required, just swap disks & NO damaging drive knock during errors or format
- Formats AS IT COPIES, detects & reproduces NON-STANDARD formats & I.D. mismatch, DOS flags
- PRODUCT SUPPORT tel. #, revisions, & upgrades

OVER 500% FASTER THAN OTHER COPIERS!

MasterCard
VISA

ONLY
$49.95
for entire system

To order CALL **805-687-1541 ext. 24**
Technical line 805-682-4000 ext. 22

AMERICAN EXPRESS

For rush service call 24 hr. order line or send cashiers check/money order. Visa, MasterCard, & American Express include card# and exp. date. CA residents add 6% tax. Allow 2 weeks for clearance of personal & co. checks. Add $3.50 shipping & handling for continental U.S./$5.50 for UPS air. AK, HI, APO, FPO, CANADA add $7.50. Other Foreign orders add $8.50. All APO, FPO, & orders outside 48 states must be paid by certified U.S. funds. C.O.D.'s shipped to U.S. addresses only, add $4.00. School orders welcome.

DISKMAKER PAYS FOR ITSELF THE FIRST TIME YOU USE IT!

Reader Service No. 112

basix

1984

3463 State Street • Suite 1541H • Santa Barbara, CA 93105

We don't care about the overall decimal value we get from the joystick port, however. All we care about, with the Commodore 64, is bits 0 through 4. Bit 0 tells us whether the joystick is pressed upward (away from the player). Bit 1 tells us whether it is pressed downward (toward the player). Bit 2 tells us whether it is pressed leftware. Bit 3 tells us whether it is pressed rightward. And bit 4 tells us whether the button is being pressed.

Since all we want is bits 0 through 4, when we read the stick we simply get rid of the other bits like this:

```
S1=PEEK(56321)AND 31:S2=PEEK(56320)AND
31
```

This automatically gets rid of all the bits that have nothing to do with the joystick.

When nothing is being pressed on the joystick, all the joystick control bits have a value of 1. That means that when the player isn't doing anything with the stick, the value of S1 and S2 in the above statements is always 31.

When the player moves the joystick, the bit that responds to that movement direction turns to 0. So if the joystick is moved left, bit 2 will be 0; if it is moved up, bit 0 will be 0; if it is moved diagonally up and left, both bit 0 and bit 2 will have a value of 0.

Commodore BASIC has thoughtfully provided us with the AND command, which lets us read each bit in turn. To read a single bit, you simply AND the variable S1 or S2 with the decimal value of that bit. For instance, the decimal value of bit 4, which reads the joystick button, is 16. Let's say your program ends if the button on joystick 2 is pressed. After executing the above statements, you could read the joystick button, or bit 4 of S2, with this statement:

```
IF (S2 AND 16)=0 THEN END
```

Remember, if the button is *not* pressed, that bit will be "on," or 1; if it *is* pressed, it will have a value of 0.

To read the right-movement bit, or bit 3, which has a value of 8, you would use:

```
IF (S1 AND 8)=0 THEN H=H-1
```

In this example, we are reading joystick 1, and subtracting 1 from the horizontal position variable H if we are supposed to move left. (If that doesn't mean anything to you, chances are you are a new *Ahoy!* reader and didn't see my earlier column on movement. Don't worry–I'll review it later in this column.)

VIC BITS

The VIC is similar to the 64 in that the individual bits tell you in which direction the joystick is being moved, and the bit will be 0 if the joystick is being moved in that direction. However, all other bets are off. That's because even though the VIC has only one joystick, the bits that report on what the joystick is doing are split between two different memory locations.

The two memory locations are 37137 and 37152.

At location 37137, bit 2 (value 4) shows upward movement; bit 3 (value 8) shows downward movement; bit 4 (value 16) shows leftward movement; and bit 5 (value 32) shows whether the button has been pressed.

At location 37152, bit 7 (value 128) shows rightward movement.

(Do not ask why the VIC was engineered to use the joystick port this way. Just remember how much you paid for your VIC and be grateful it has a port at all.)

To combine all the pertinent bits into one variable, S, this statement does the job:

```
S=(PEEK(37137)AND 60)OR(PEEK(37152)AND 1
28)
```

After this statement, the variable S will have all the information, so that we can check each direction of movement with the following statements:

```
IF (S AND 4)=0 THEN V=V-1
IF (S AND 8)=0 THEN V=V+1
IF (S AND 16)=0 THEN H=H-1
IF (S AND 128)=0 THEN H=H+1
```

(The V variable controls vertical position, the H variable controls horizontal position. To move up or left, subtract 1; to move down or right, add 1.)

To read the joystick button and jump to a subroutine at line 900 if it is pressed, we'd use this statement:

```
IF (S AND 32)=0 THEN GOSUB 900
```

HERE COME THE PROBLEMS

In one of its occasional strokes of unbelievable genius, Commodore's design department came up with the clever idea of having the keyboard and the joystick read through the same locations in memory. This affects both the VIC and the 64. With the 64, however, only joystick port 1 is affected–joystick port 2, the one next to the on-off switch, is free and clear. That's why so many games–and the demonstration program that comes with this article–use only joystick port 2. Why add unnecessary grief?

But you may want to program using two joysticks with the 64, and with the VIC you have no choice at all–you have to get around the keyboard problem.

The solution is, of course, to disable all or part of the keyboard. This means that before reading the VIC joystick, your VIC program must

```
POKE 37154,127
```

Then, to reenable the keyboard, you must

```
POKE 37154,255
```

And before reading the 64's joystick 1, your 64 program must

```
POKE 56333,127:POKE 56320,255
```

and reenable the keyboard with

```
POKE 56333,129
```

You can disable the keyboard at the beginning of your program and reenable it at the end, *if* you're playing joystick-only games; but what if you wanted to use some keys during the game, too? Like the function keys, for instance?

In that case, you have to disable the keyboard just before reading the joystick, and reenable it immediately afterward, *every time* you read the joystick, and then read the keyboard somewhere else in the main loop. Since POKEs are notoriously slow, adding two or three of them to every pass through the movement loop can slow down your program considerably.

This is not a problem for machine language programmers, of course, since machine language is so very fast—and nothing is simpler than LDA #127 followed by STA 56333. But for you and me, programming games in BASIC at home, it can be quite annoying.

CUTTING CORNERS

Nobody told you about that before you bought the computer, did they? Lots of talk about RAM and ROM and sprites and price and amounts of software, but nobody ever said, when it comes to *programming* using the joystick and the sprites, well, you're on your own, and we haven't exactly made it easy or fast to do it in BASIC.

Of course they didn't. For one thing, if your VIC or 64 was your first computer, chances are you wouldn't have known what they were talking about. The people designing the computer dealt with programming at an engineering or operating system level—these things didn't matter to them. And the people designing BASIC moved the old CBM BASIC from PET computers to the VIC and 64 without adding any routines to handle graphics or joysticks. And why? To save nickels and dimes, of course, so we'd see the lower price and buy the machine by the millions.

But now you know, don't you? Now you've learned BASIC programming, and you realize that when they cut a corner to shave 98 cents off the price, you pay for it in slower and harder-to-write programs.

GRIPE, GRIPE, GRIPE

Don't misunderstand me, please. I'm not saying they didn't sell us a great little computer. Twenty years ago, computers as powerful as the VIC and the 64 took up whole rooms. You've got a great computer for less than your TV cost you.

Furthermore, it's not as if nobody *else* cut corners. The VIC and the 64 have problems, but so do all the other home computers. Atari has the famous bug in its BASIC that can wipe out whole programs at a single bound, and its DOS is a horrible timewaster; the Adam is famous for being able to destroy the entire BASIC cassette if you happen to turn the machine off or on with the cassette anywhere close by; the PCjr has unbelievable eccentric screen memory arrangements, no sprites, and a very strange and hard-to-program joystick; the Apple is a relic of the Dark Ages and can't do anything useful with graphics or sound in BASIC—or, to tell the truth, in machine language either.

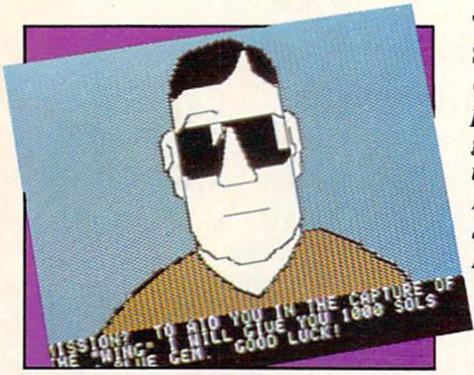
NEXT TIME, LET'S GET IT RIGHT

However, just because nobody's perfect doesn't mean nobody *can* be. Commodore has worked out an arrangement with Amiga to produce what is rumored to be the best home computer ever. I have heard so many stories about this marvelous machine that I halfway expect it to be announced by choirs of angels. The skeptical side of me says that the same people who brought up great graphics with no BASIC implementation are perfectly capable of doing it again. But I have an optimistic side, too.

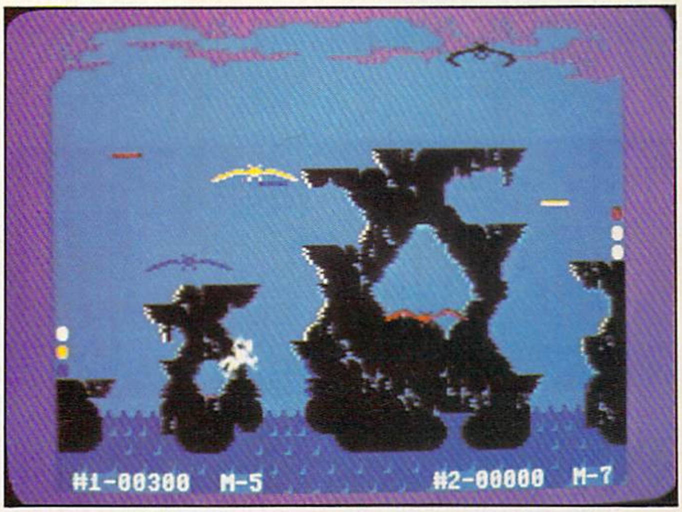
So let me describe to you what the perfect home computer will be. The new computer that is worth buying even if you already own a Commodore 64—and, to tell you the truth, *no* home computer on the market at the moment I'm writing this (January 1985) is worth buying if you already have a 64. Only if you want to jump away to business machines that largely ignore the computer as an entertainment and hobby medium is your 64 not perfectly adequate.

But if Commodore actually comes out with the dream machine, then yes, all the other home computers in my

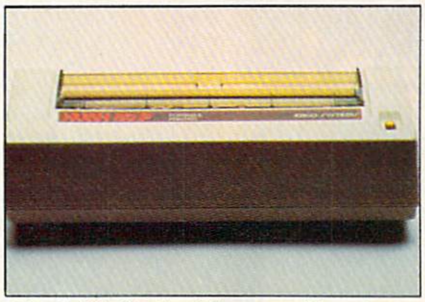
Continued on page 61



The Tracer Sanction ***(left) is a patchwork of good and bad ideas. READER SERVICE NO. 105***



Indiana Jones ***(p. 28) must survive six danger-filled rooms. READER SERVICE NO. 106***



Hush 80 CD (p. 25): portable printer. READER SERVICE NO. 107

THE TRACER SANCTION

Activision
Commodore 64
Disk; $34.95

An interstellar archfiend known only as The Wing is on the loose, and it's up to the player to bring him to justice. In this illustrated adventure, the spacegoing sleuth travels from planet to planet in a one-seat cruiser to collect clues leading to the ultimate confrontation with the master criminal.

The Tracer Sanction has significantly more text than most previous illustrated adventures. Although there is a drawing for every major location in the game, the accompanying text generally runs to several paragraphs, instead of just a line or two. This allows the authors to provide much fuller descriptions of places and events and even work in the kind of scene-setting and characterization which adventure gamers have previously expected only from all-text programs.

The artwork varies in quality. The illustrations actually get better as the story unreels, indicating a mid-development switch to a different graphics system. (*Mindshadow,* the companion game to *The Tracer Sanction*, uses the better grade of drawings throughout.)

Minor animations dress up several of the pictures. For instance, when the protagonist lines up to use a information computer, the gamer can see images flickering across the terminal's screen. Another attractive feature is that the program puts the drawings on the screen very rapidly. This is one game in which you won't get clues from watching a too-slow fill.

The stymied player can obtain some assistance, however, by typing in the word "Condor." This summons a highly intelligent avian. The Condor stands—or should that be flies?—apart from the game, and is frequently busy with plans and problems of his own. Generally, the wily bird's comments on his situation suggest a way out of the adventurer's predicament as well. The rules limit the gamer to three hints.

Typical of the program's technical virtuosity is the "save game" feature. Ten different positions can be saved to disk for later recall, while the "quicksave" option lets the player consolidate gains made in the game-in-progress with barely an interruption in the flow of play.

The Tracer Sanction is a computer novel in that events proceed in linear fashion with little room for deviation. This is acceptable in a futuristic mystery like this in which the detective is expected to go from clue to clue in logical progression. Common sense and keen observation are the main requirements for successfully completing the game, and the program is mercifully free from the kind of intricate, abstruse puzzles which frustrate more than they challenge.

The plot is a patchwork of good and bad ideas. *The Tracer Sanction* betrays its illustrated adventure lineage in annoying little ways, such as the fact that the detective must pick up everything that isn't bolted down in the sure knowledge that it will be useful sometime. As is generally the case in such games, the computerist has to die a few times while learning that the seemingly useless object lying around on one planet is the key to success at a later stage.

Explaining the really clever scenes would only ruin them. Suffice to say that adventurers should always bear in mind the writers' barbed sense of humor when faced with a seemingly hopeless situation.

The negative aspects of *The Tracer*

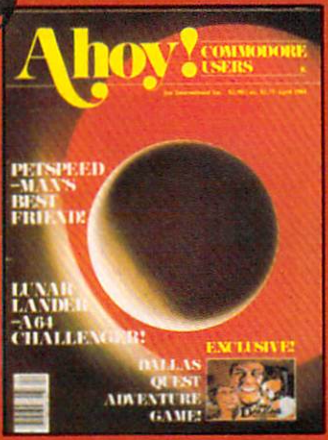
Back Issues of Ahoy!





Sorry—issues #1 and #2 sold out! Limited edition reprints (programs and feature articles only!) available for $4.00 each.





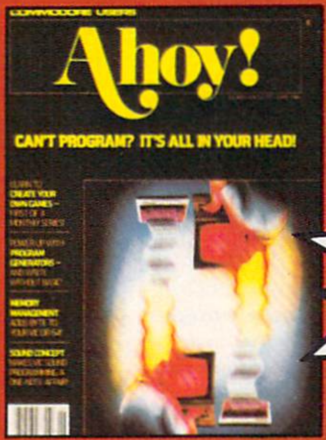
ISSUE #1–JAN. '84 $4.00
64 v. the Peanut! Telecommunications! Tales of the Commodore! And ready to enter: Multi Draw 64 Graphics System! Interrupt Music Maker/Editor! A Peek at Memory! Sequential Files!

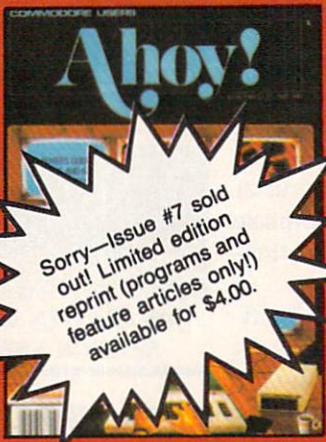
ISSUE #2–FEB. '84 $4.00
Illustrated tour of the 1541! Artificial intelligence! Synapse's Ihor Wolosenko interviewed! And ready to enter: Music Maker Part II! Screen Manipulation! Night Attack! Relative Files!

ISSUE #3–MAR. '84 $4.00
Anatomy of the 64! Printer interfacing for VIC & 64! Educational software series begins! And ready to enter: Address Book! Space Lanes! Random Files on the 64! Dynamic Power!

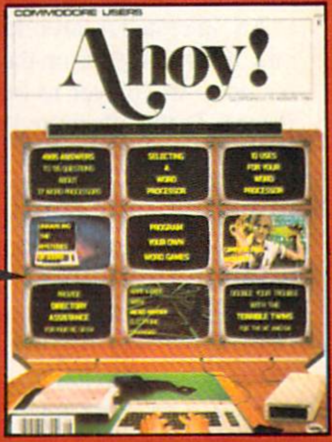
ISSUE #4–APR. '84 $4.00
Petspeed and Easy Script tutorials! Printer interfacing continues! Laserdisc! And ready to enter: Apple Pie! Lunar Lander! Name that Star! Lower case descenders on the 1525!







Sorry—Issue #7 sold out! Limited edition reprint (programs and feature articles only!) available for $4.00.



ISSUE #5–MAY '84 $4.00
Future of Commodore! Inside BASIC storage! Memory management on the VIC & 64! Spreadsheets! And ready to enter: Math Master! Air Assault! Biorhythms! VIC Calculator!

ISSUE #6–JUNE '84 $4.00
Game programming column begins! Program generators! Rupert on inputting! Memory management continues! And ready to enter: Post Time for the 64 & VIC! Alpiner! Sound Concept!

ISSUE #7–JULY '84 $4.00
MSD Dual Disk! Database buyer's guide! Training your cursor! Screen displays! File Sleuth! Users Groups! And ready to enter: Renumbering! Checklist! Math Defender! Brisk!

ISSUE #8–AUG. '84 $4.00
Choosing a word processor! Computational wizardry! Creating your own word games! Sound on the 64! And ready to enter: Micro-Minder! Directory Assistance! Terrible Twins!

ISSUE #9–SEPT. '84 $4.00
Program your own text adventure! Build a C-64 cassette interface! Video RAM! And ready to enter: Salvage Diver! DOS! Sound Explorer! Castle of Darkness! Base Conversions!

ISSUE #10–OCT. '84 $4.00
C-64 Graphics programs! Bit-mapped graphics! Joystick programming! And ready to enter: VIC 40 Operating System! BAM Read & Print! Emerald Elephant of Cipangu! Lawn Job!

ISSUE #11–NOV. '84 $4.00
Music programs & keyboards for the 64! Graphics feature continues! And ready to enter: PTE word processor! Block Editor! Alternate Character Set for the 64! Tunnel of Tomachon!

ISSUE #12–DEC. '84 $4.00
Buyer's guide to printers! 1525 printer tutorial! Custom characters! User Guide to KMMM Pascal! Diving into BASIC! And ready to enter: Construction Co.! Space Patrol! Cross Ref!

ISSUE #13–JAN. '85 $4.00
VIC/64 OS exposed! Sprites! 1541 device # disconnect switch! Ghostbusters! And ready to enter: Ultra Mail! Music Tutor! Alice in Adventureland! Midprint! To the Top!

ISSUE #14–FEB. '85 $4.00
Printer interfacing! Multicolor sprites! Modems! Bulletin boards! Theory of game design! And ready to enter: Futurewar! Fontasia! VIC Eraser! Insurance Agent! Telelink 64!

ISSUE #15–MAR. '85 $4.00
Making multiscreen gameboards! Inside the Plus/4! Commodore DOS! And ready to enter: Old Routine! Programmable Functions! Automatic Line Nos.! Home Budget! Salmon Run!

If you have a modem and want more complete information on any of the issues listed, call the *Ahoy!* Bulletin Board: 212-564-7727.

Send coupon or facsimile to: *Ahoy!* Back Issues, Ion International Inc., 45 West 34th Street—Suite 407, New York, NY 10001.

Ahoy!

Please Send Me The Following:
____ Copies of issue number ____
____ Copies of issue number ____
____ Copies of issue number ____

Enclosed Please Find My Check or Money Order for $____

(Outside the USA please add $1.00 for every copy)

NAME ____
ADDRESS ____
CITY ____
STATE ____
ZIP CODE ____

Sanction notwithstanding, it is a pleasure to play. It moves along briskly with few dead spots and can be solved in 30-40 hours at the keyboard.

Activision, Inc., 2350 Bayshore Frontage Road, Mountain View, CA 94043 (phone: 415-960-0410).

–Arnie Katz

HUSH 80 CD
Ergo Systems, Inc.
C-64, VIC 20; $140
(See photo on page 23)

In brief, the Hush 80 CD is an 80-column thermal dot-matrix printer which is fully compatible with the C-64 and VIC 20. It is extremely compact, lightweight (22 ounces) and quiet in operation, making it eminently suitable for those late night sessions in a remote motel room. Overall dimensions are 11.625 by 5.5 by 2.8 inches, which includes a roll of paper inside the printer housing. If you do intend to travel, be sure to take into account the weight of the paper (11 ounces for a full roll) and the external power transformer (about one pound). Nevertheless, you will still end up with a less than three pound package which will easily fit in an attaché case or a commodius shoulder tote.

The printer, which comes with an integral Commodore serial port cable, has a built-in interface without any serial port connectors. It is permanently configured as device number four. All the 1525 commands, with two exceptions, were fully implemented. (See the December 1984 *Ahoy!* for a complete tutorial on the features and operation of the Commodore 1525 printer.) The first roll of printer paper is preinstalled, which is a very good thing, as loading the paper is not a task for the fumble fingered. Manual paper advance is by a pushbutton on top of the printer.

We noticed two oddities when running the Commodore printer test. After operating in cursor down mode, the printer did not revert back to cursor up mode when the channel was reopened without a secondary address. The default zero must be explicitly stated: for example, OPEN 1,4,0. Commodore documentation specifically states that omitting the secondary address defaults to zero. The printer also remained in reverse print mode after the completion of the test. An Ergo spokesperson indicated that these were deliberate modifications to make operation "easier." We feel that these changes could cause difficulties with some existing software.

Overall operation was good, although the print quality was only fair. Since this is a thermal printer, you must use that metallic-feeling heat sensitive paper available only in rolls. The friction feed was firm, nonadjustable, and limited to a paper width of 8½ inches. The full Commodore character set is supported in a 5 x 7 dot matrix without lower case descenders. Graphic characters are printed in a 6 x 7 dot matrix. These formats are identical to the 1525. The actual character width was less than ten characters per inch. The default printing pitch was 12.8 characters per inch. Line spacing was a standard six lines per inch.

Printing speed is specified at 80 characters per second bidirectional. The printer seems to be fitted with a 90 character buffer, as is the Commodore 1525. It took the Commodore printer test program six minutes and eighteen seconds to run to completion. A double size high resolution *DOODLE!* dump took ten minutes and 44 seconds, and a single size dump required two minutes and twenty-seven seconds.

High resolution graphics printed very well with the Hush 80 CD. Dot spacing and density were extremely uniform. Since the paper roll is entirely contained in the printer case, very uniform feed drag results. This, combined with the very tight paper feed mechanism, eliminated all linefeed gaps in the high resolution image. The narrow dot size resulted in very well proportioned horizontally formatted images. However, rotated images such as the double size *DOODLE!* dump came out rather elongated.

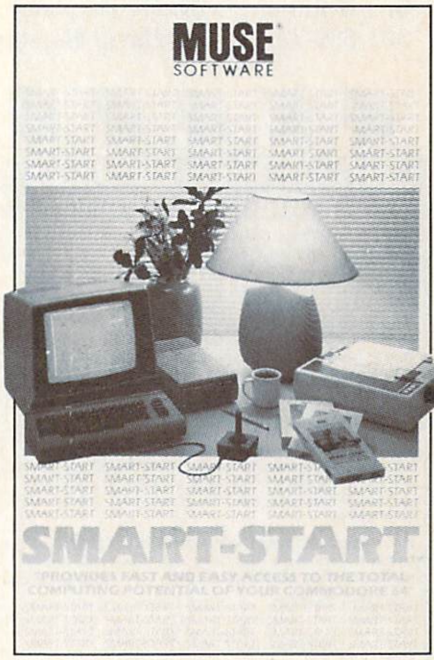
Overall, the Hush 80 CD is a good low cost first printer or a handy travel printer where size and weight are important. The fair print quality on thermal paper makes it suitable only for less aesthetically demanding tasks. Bear in mind that the $140 price tag is on the order of the cost of a good Commodore printer interface without a printer. Serial (RS232) and Centronics versions of the printer are also available.

Ergo Systems, Inc., 26254 Eden Landing Road, Hayward, CA 94545.

–Morton Kevelson

SMART-START
Muse Software
Commodore 64
Disk; $39.95

Most new and prospective Commodore 64 owners are enthused to learn of the machine's powerful sound and sprite capabilities. Their exuberance fades quickly once they learn of the difficulties involved in programming sound or sprites.



READER SERVICE NO. 104

Smart-Start overcomes this problem by providing simple editors that allow beginners to work with sound and graphics. Afterwards, *Smart-Start* will even generate the appropriate program lines.

In addition to providing these aids,

Smart-Start places a comfortable programming environment into the C-64, entered or exited by the stroke of RUN STOP/RESTORE. Once there, you are faced with a screenful of information and choices including a digital clock, alarm, memory available (on disk and in RAM), LOAD, RUN, SAVE, COLORS, and UTILITIES. Through the UTILITIES option you can set the alarm or clock, print the screen or a program, scratch files, format a disk, or enter the sprite and sound editors. Once you exit, you may use your C-64 as normal.

Smart-Start is what the legions of new Commodore owners need—a carefully conceived, easy to use utility that offers plenty of aid for the money.

Muse Software, 347 N. Charles St., Baltimore, MD 21201 (phone: 301-659-7212). *—David Barron*

MERLIN 64

Roger Wagner Publishing, Inc.
Commodore 64
Disk; $49.95

(Though not a newcomer to the marketplace, we provide this review of *Merlin 64* for the benefit of those readying to purchase their first assembler for use with our new series on assembly language [see *Commodore Roots,* page 51]—Ed.)

Providing all the features of expensive assemblers, the *Merlin 64 Macro Assembler* is an excellent value. It requires a C-64 with disk drive, and can be used with the DATA20 Videopack80 column card. It also works with the Commodore RS232 cartridge.

Merlin creates executable program files that can be loaded and run without modification. Number values (including memory addresses) can be entered in decimal, hex, or binary. Macros can be used to do repetitious tasks.

The manual assumes you know how to use an assembler and doesn't attempt to teach programming. Wherever appropriate, it takes a step-by-step approach and uses sample programs. A technical section provides details needed to tailor the program to individual preferences. A glossary and error messages section round out the manual.

The design of *Merlin 64* suggests that the author intended it for his own use. The convenience features are too well formulated to have been created by someone who doesn't do a lot of programming. For instance, there are two different SAVEs: one for source code, another for object code. The object code SAVE only works after a successful assembly.

The program stashes the current filename and uses it as a default when a disk save is done, unless a new name is entered. The source code has a .s automatically appended to the name; object code gets a .o tacked on. A drive specifier can be added to write the file to a different disk.

Of course, there is a LOAD file option, but *Merlin* goes one step further. A read command puts text files into *Merlin.* This allows you to use source code edited with a word processing program that creates sequential files. It also makes it easy to enter source code that has been downloaded. A write command does the reverse: creates a sequential file of source code. A utility program provided on the disk converts and loads source code created with Commodore's assembly language system.

Other disk options let you view a catalog of the files on the disk or scratch any of them. An append command makes it easy to add sections of source code to the end of the current program.

An important aspect of any assembler is its editor. Since the four fields used by an assembler (labels, opcodes, operands, and comment) are all limited to a certain number of spaces, the editor can use tab stops to make entry simple. *Merlin* handles this by having automatic tab stops. The tabs can be changed, but the default setting is perfect for most uses.

Merlin's editor is similar to the C-64's BASIC editor operating in AUTO mode, but adds many word processor features: insert, delete, find, copy, move, change. These commands can affect words, strings of characters, lines, or ranges of lines

A print function allows you to send source code out ports 2, 4, or 5, or to the screen. Files may be printed in two formats: line by line as they appear onscreen or in page format with headers, page numbers, and page breaks. Pretty snazzy for an assembly program.

The assembler can be accessed from the disk functions mode or from the editor. A real convenience feature is the update option just before assembly begins. By choosing "yes" at the prompt, the assembler finds the first line containing a "/" and goes into edit mode. By entering a date or other identifying notation, you can be sure which is the latest version of source code that's been assembled. Keeping track of your latest update can be a real hassle when you are busily modifying code.

There are several pseudo-opcodes that add flexibility to the program. By using three of these, PUT, SAV,

and DSK, very large programs can be assembled. PUT reads in text files to be assembled, SAV writes the object code to disk, and DSK directs the assembler to output the code directly to disk.

Merlin also has conditional assembly. This feature allows for slight changes in a program without having to duplicate the whole thing. Using pseudo-opcodes like IF and ELSE portions of code can be ignored for different assemblies.

But *Merlin* doesn't stop there. It not only assembles but also disassembles code. This is one feature that really makes this program stand out. By studying code written by others, new programming techniques can be developed and good routines can be borrowed. A special label feature will identify any memory location function that it recognizes: for instance, jumps into the Kernal.

Merlin lets you run an assembled program from inside itself, so you can check how well your assembled code works. Since *Merlin* loads itself into an out-of-the-way corner, most of BASIC memory is available for your programs. Exiting from *Merlin* doesn't wipe it out. A simple SYS command puts you back into it.

Merlin 64 is an excellent little assembler with many value added features. For ease of use, I can't imagine how it could be better. The utilities that *Merlin* gives you include a text formatter, an 80-column converter, a cross referencer for labels, and a stripper that removes all comments. A group of full-commented, commonly used routines is included to simplify ordinary tasks. The author has gone out of his way to add anything he can to make the program an outstanding value.

Roger Wagner Publishing, Inc., 10761 Woodside Ave., Suite E, Santee, CA 92071 (phone: 619-562-3221). *–Cheryl Peterson*

MUSIC PORT
Tech-Sketch, Inc.
Commodore 64
Keyboard and disk, $119.95

In the past few months the marketplace has been inundated with musical keyboards for the Commodore 64. Tech-Sketch's Music Port is a competitive addition to the lot.



Music Port adds non-SID sound capabilities to the 64 through software.
READER SERVICE NO. 125

The Keyboard. A full three octaves housed in a sturdy tan case. All the keys are full size and full travel. The keys are also very responsive, making this one of the most playable keyboards on the market.

The Software. The included disk contains one program that serves as the keyboard's control center. Additionally, you get sample songs, sequences (bass lines you can solo over), and many different sounds. My preview copy had 64 sounds, though I'm told the final version will have more. Overall, the software is excellent: the package included allows you to play live, multitrack songs, and define sounds. An extra $30 buys you a light pen and light pen-interactive composition software, not tested for this report.

Sound Definition. Probably the strongest point of the package. When in this mode you are presented with a hi-res screen that more closely resembles the front panel of a synthesizer than anything else. This is because Bruce Brody (professional musician and co-author) has placed features into the control panel that are not supported by the SID chip (pro-

duced through software). The first major addition is an LFO (low frequency oscillator). This generates a waveform (sine, square, ramp, random, and others are supported) at low frequencies that can be linked into the frequency, filter, or pulse width of any or all of the oscillators. Resulting are vibrato, resonant sweeps, and other electronic effects.

A unison mode is also supported. When unison is selected all three oscillators work on producing one sound. The result is a layered "fat" sound that is useful in creating certain sonic effects. All in all, the sound selections are superb, ranging from percussive organs to Star Trek phaser noises to resonant sweeped orchestras.

Multitracking. If you thought multitracking at home meant spending megabucks on fancy tape machines, think again. With Music Port you can lay down three tracks of music on top of one another when creating a composition. Each track corresponds to one of the three SID oscillators. Each track can also be a separate instrument, allowing you to lay down a bass line on one track, an organ on another, and finally a flute on the third. When recording a track the previous tracks are played, and if you wish, a metronome can be activated to help you with timing. You may also elect to record all three tracks at once.

Use. The entire software package is menu-driven and controlled from three keys on the musical keyboard. Two consecutive keys move the pointer up and down through the menus while one of the end keys serves as an ENTER key. The only time you are forced to go back to the Commodore's keyboard is to enter a filename to SAVE or LOAD.

The vast array of features and power of the sound section make the Tech-Sketch Music Port an excellent choice for amateur and professional musicians alike.

Tech-Sketch, Inc., 26 Just Road, Fairfield, NJ 07006 (in NJ 201-227-7724; rest of USA 1-800-526-2514).

–David Barron

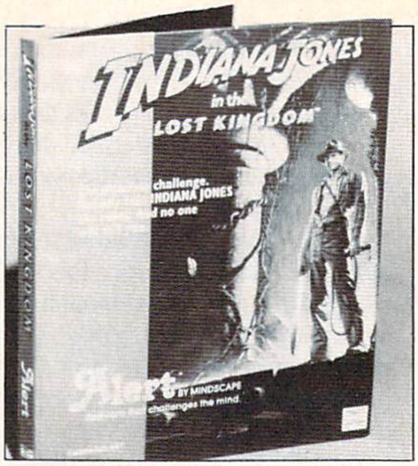
INDIANA JONES IN THE LOST KINGDOM

Alert by Mindscape
Commodore 64
Disk; $29.95

(See screen on page 23.)

In combining the rigors of arcade style hand/eye coordination with elements of interactive strategy, the programmer of *Indiana Jones in the Lost Kingdom* tried to provide a game that would remain enthralling after many hours of play. In this goal he succeeded. But until those hours have been logged, *Indiana Jones* is not enthralling–just maddening. No instructions have been provided, and your first few sittings will be spent figuring out the rules.

You're not left to fly totally blind. You're told this much: famed archaeologist Indiana Jones must find an artifact of a lost kingdom worth an incalculable fortune. Armed with only a mystical cane, Indy must wend his way through six hazard-filled rooms and find the treasure ahead of his arch-rival Ivar Reiss.



Indiana Jones faces a host of perils in this arcade/strategy game.

Perils include vampire bats, carnivorous monsters, killer snowflakes, twisting mazes, and treacherous cliffs. A set of clues is provided for each room, along with a hint hotline phone number should you really get stuck.

While the clues are straightforward enough, it will take even videogame *virtuosi* some time to figure out the action in each room. One setting requires you to scale steep cliffs while avoiding bats; another sends you cavorting around mountain peaks dodging killer snowflakes (I wasn't kidding about those); a third involves an underground maze with twisting passageways and hidden horrors. Joystick movement differs in each and will require you to "relearn" how to play.

I won't reveal the rules except to say that once the logic of gameplay has been determined, it's easy enough to discover what's required in each room. The trick is figuring out how to perform the various tasks required without killing all seven of your men. Fortunately in this regard, you're able to enter any room at any of the three levels without having to pass through the previous one.

I must admit, the lure of the Lost Kingdom is strong, but having to fumble through several sessions without direction almost sent me hying back to civilization. No documentation may have sounded good in the planning stages, but at the

Continued on page 82

Disk Spinners

Replacements and Enhancements for the Commodore 1541

By Morton Kevelson

This month we examine the Indus GT and Commander II drives, 1541 Physical Exam, and On Track Indicator. Next month we continue with Kwik-Load!, Fast Load, and 1541 Flash.

PART I

DISK DRIVES AND OTHER GIMMICKS

It is interesting to note how good products are often imitated, usually by lower cost competitors. We have all seen this in the microcomputer world, most notably with the flock of IBM PC lookalikes presently available. Interestingly enough, no one makes a low cost lookalike for the Commodore 64. This tells us something about the price and performance of the computer which many of us have chosen as our own.

When it comes to the matter of major peripherals for the C-64, there seem to be many who have something to offer. Recently, it is the 1541 disk drive which has come under attack. Invariably, these are not lower cost alternatives. The price of the 1541 disk is on the order of $200 in the New York City area as of this writing. By the time you read this, it may be even lower.

What all these competitors offer is improved performance—most notably in the area of speed. This is not normally possible. The time required to perform the basic operations of LOADing and SAVEing programs is primarily in the hands of the computer. The same is true for reading and writing data files. Certain disk operations, such as formatting, can be improved upon. But let's face it: just how much time does the average user spend on formatting disks?

This is not to say that improved speed is not possible. On the contrary, a quick look at the accompanying reviews will show that speed improvements of better than five to one have been achieved. In all cases, without exception, this speed improvement is accomplished by a change in the software which is responsible for controlling both the computer and the disk drive. In some cases, an accompanying change in the hardware is required as well.

As a potential purchaser of a non-Commodore disk drive or of a disk drive enhancement product, you will have one other major concern: compatibility. The problems with compatibility are subtle. They make take some time to manifest themselves, very often in unexpected ways. An excellent example is the destructive write incompatibility between the Commodore 4040 and the 1541 disk drives. The specific details of this have been well-documented. We are reasonably certain that problems of this sort will not crop up in the disk drives being offered as replacements for the 1541.

There are two levels of compatibility which the user must be concerned with. The first is compatibility with the Commodore DOS commands. Since these have all been well-documented at this time, we do not expect to find any problems in this area. The second level deals with commercial software and the concept of copy protection.



Commander II: *1541 compatibility.*
READER SERVICE NO. 101



Indus GT: *utilities stored in ROM.*
READER SERVICE NO. 102

The Commodore disk drive is an intelligent peripheral. It is in effect an independent computer with quite literally, a mind of its own. It comes complete with its own microprocessor, a 6502—identical to the one in the VIC 20 and closely related to the 6510 used in the C-64. It has built-in RAM—two kilobytes of the stuff. It is complete with its own input/output (I/O) hardware—a pair of 6522 versatile interface adapter chips (VIA). Unlike with the C-64, the I/O channels are not humanly understandable. Instead of a keyboard and video monitor, the disk drive communicates with the computer via the serial bus on the one end and the

drive mechanism motors on the other end.

The built-in controlling package, the DOS, is a sixteen kilobyte machine language program which is stored in two eight kilobyte ROM chips inside the 1541. This program is the equivalent of BASIC and the Kernal in your computer.

Like the computer, the user may tell the DOS what to do next. Unlike the computer, the disk drive has no keyboard for user interaction. Instead, all instructions are passed along via the serial bus. Like the computer, the disk drive has some built-in RAM for storing data and additional machine language instructions. This memory is rather limited, just two kilobytes for the 1541. Also like the computer, there is a DOS command which is similar to BASIC's SYS command. This is the MEMORY-EXECUTE command. Clever commercial programmers have used these features to make the disk drive behave in totally unexpected ways. This forms the basis for commercial copy protection schemes.

These protection methods will usually involve direct execution of machine language routines which are stored in the DOS ROMs. This is where the compatibility problems crop up. Unlike the computer, the DOS has no equivalent to the Kernal routines. Commodore makes no guarantees about the internal organization of the DOS. In addition, the copyright laws prohibit the exact duplication of the 1541 DOS by unlicensed manufacturers. As a result there are no guarantees that the clever routines which are being used will be the same in somebody else's disk drive. For that matter, there is no guarantee that future versions of the 1541 DOS from Commodore will be the same.

Fortunately, things are not all that bad. Commodore seems to be making some efforts to keep changes in the DOS to a minimum. The only problem that remains is if a software producer chooses a protection scheme which happens to involve portions of the Commodore DOS that are just not the same in the non-Commodore drive. Even in this regard, things are looking up. Both the disk drive makers and the software producers have recognized the need to maintain compatibility to avoid limiting their own markets.

ON SPEED

The simplest way to improve the speed of disk operations is to indirectly modify the machine language controlling routines buried in the Kernal and the DOS. This of course requires the storage and some additional code somewhere in the computer's RAM. This is just the approach used by Indus and Datamost with their products. The problem, of course, is that if you can find a tidy spot in RAM for the speedup software, someone else is apt to covet the same location for their own purposes. This is just what actually happens to limit the performance of this approach.

The other approach is to replace the code in firmware, as with the Skyles *Flash*. This raises the problems of possible built-in incompatibility with some software or hardware problems.

What it all boils down to is that no matter what approach you take, there will be some compatibility risk involved. The ultimate decision lies in the hands of the user. ☐

THE COMMANDER II DISK DRIVE

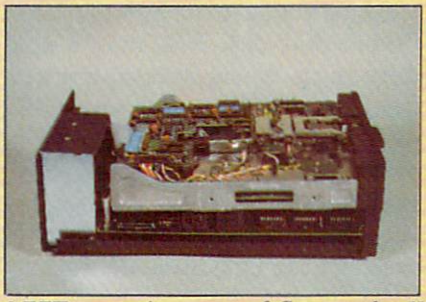
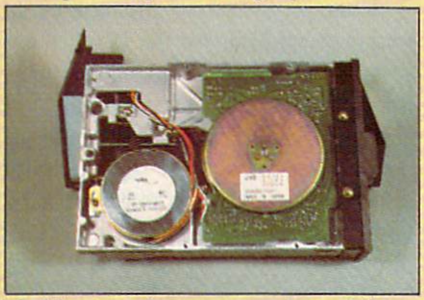
Commander Electronic Systems
P.O. Box 15485
Santa Ana, CA 92705
Phone: 714-953-6166
Price: $289.00

Upon examining this disk drive, the first question that comes to mind is: what happened to Commander I? The model II is your plain, no frills disk drive. Its compact 11 by 6 by 3½ inch steel package makes it only 52 percent of the volume of a 1541. This is a true size reduction, as the power supply is built in.

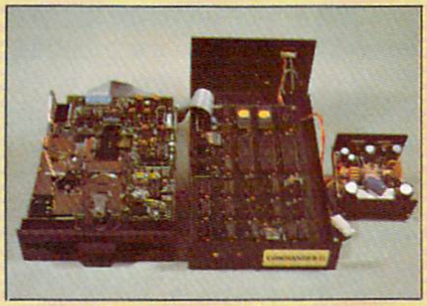
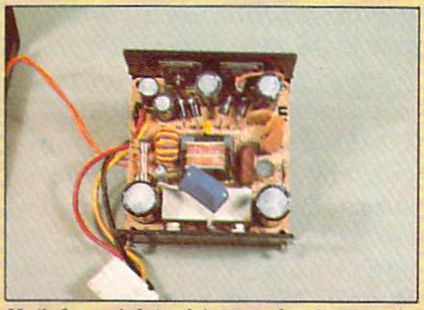
Operating speed is basically the same as for the 1541 with a 75 second disk formatting time—a slight improvement. Compatibility is very high, giving us no trouble with any of the software we tried.

Physical construction is a bit unusual. The twin serial port connectors are on the left side of the drive as opposed to the traditional back location. The power indicator light on the front panel is red, while the drive activity light is green—the opposite of the 1541. A separate red indicator light tells you when the drive mechanism itself is active.

A pair of protruding lips flank the

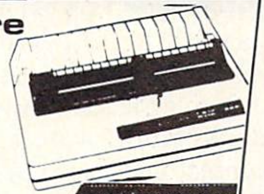
LEFT: *removing cover of Commander II disk drive reveals power supply module on left, drive mechanism on top, and main circuit board on bottom.* **RIGHT:** *bottom of drive, left to right: stepper motor, direct drive disk motor manufactured by JVC.*

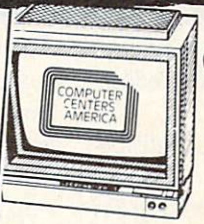
LEFT: *three subassemblies of Commander II (left to right): drive mechanism, main circuit board, and power supply module with cover removed.* **RIGHT:** *closeup of the power supply module. The hidden fuse is to the left.*

WE'RE PROUD OF OUR PRICES!

commodore
NEW
1101 LETTER QUALITY PRINTER
274^{95}

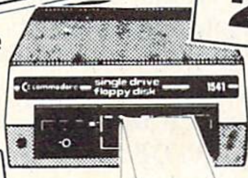


NEW
commodore
128
IN STOCK
279^{95}

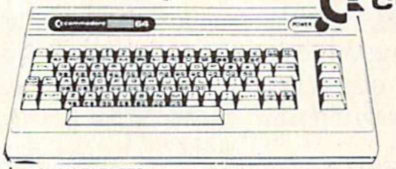


commodore
1702 MONITOR
174^{95}

commodore
1541 DISC DRIVE
159^{95}



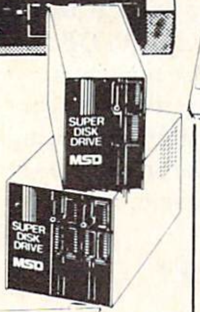
commodore 64
149^{95}

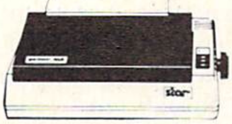


ALL CARDCO PRODUCTS IN STOCK & ON SALE!

NEW! MSD SUPER DISC II
NOW ON SALE
459^{95}
Single Drive 249^{00}







Modems

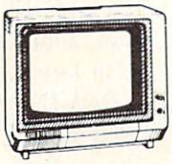
Comm 1660	$99
Mighty Mo	Call
Telesonic	Call
Hes II	Call

Interfaces

Cardco G +	Call
Microworld	Call

Cassette Recorders

Comm CN-2	59^{95}
Third Party	39^{95}



Monitors

BMC Color	169^{95}
Pan 1300 RGB & Comp.	289^{95}
BMC Green	69^{95}

Printers

Gemini SG 10/15 New!	209^{95}
Gemini SD 10/15 New!	Call
Gemini SR 10/15 New!	Call
Gemini SB 10 New!	Call
Gemini 15X	399^{95}
Silver Reed 400	249^{95}
Silver Reed 500	379^{95}
Brother 15X	379^{95}
Brother 25X	514^{95}
Juki 6100	399^{95}
Blue Chip with Interface	219^{95}

Diskettes

BASF	13^{95}
Scotch	14^{95}
Maxell	14^{95}
Polaroid	13^{95}

TOUCH TABLETS	
Koala Touch Tablet–D	64.95
Koala Touch Tablet–Cart	69.95
Muppet Learning Keys	Call
ACCESS	
Neutral Zone–D T	20.95
Spritemaster–D T	20.95
Beachhead-D/T	20.95
Master Composer–D	23.95
SPINNAKER	
Adventure Creator–Cart	21.95
Aerobics–D	25.95
Aegean Voyage–Cart	21.95
All in the Color Caves–C	21.95
Alphabet Zoo–Cart	20.95
Bubble Burst–Cart	21.95
Cosmic Life–Cart	20.95
Delta Drawing–Cart	21.95
Facemaker–Cart	20.95
Fraction Fever–Cart	20.95
Grandma's House–D	20.95
Jukebox–Cart	20.95
Kids on Keys–Cart	20.95
Kidwriter–D	20.95
Kindercomp–Cart	17.95
Ranch–Cart	20.95
Rhymes Riddles–D	17.95
Search Amazing Thing–D	24.95
Snooper #1–D	24.95
Snooper #2–D	24.95
Story Machine–Cart	20.95
Trains–D	24.95
Up For Grabs–Cart	20.95
COMMODORE	
Program Ref. Guide	19.95
Assembler–D	17.95
Easy Finance I,II,III,IV–D	19.95
Easy Calc–D	64.95
Easy Mail–D	17.95
Easy Script–D	39.95
Easy Spell–D	19.95
Logo–D	49.95
The Manager–D	37.95

NEW!

commodore
Products

801 MCS Color Printer	269^{95}
C-16 Computer	79^{95}
C+4 Computer	189^{95}
801 Printer	149^{95}
802 Printer	179^{95}
803 Printer	174^{95}

General Ledger–D	37.95
Acct. Rec–D	37.95
Accts. Pay–D	37.95
Magic Desk–D	52.95
Zork I, II or III–D	29.95
Suspended–D	29.95
Starcross–D	29.95
Deadline–D	29.95
CBS SOFTWARE	
Argos Expedition–D	29.95
Charles Goren's Bridge–D	49.95
Coco Notes–D	19.95
Ducks Ahoy–D	23.95
Ernie's Magic Shapes–D	23.95
Mastering the SAT–D	104.95
Movie Musical Madness–D	24.95
Murder by the Dozen–D	23.95
Peanut Butter Panic–D	24.95
Sea Horse Hide 'n Seek	24.95
Success Decimals (Add Subt)–D T	14.95
Success Decimals (Mult Div)–D T	14.95
Success Fractions (Add Subt)–D T	14.95
Success Fractions (Mult Div)–D T	14.95
Timebound–D	24.95
Webster Word Game–D	24.95
ACCESSORIES	
WICO Joysticks	Call
Flip 'n' File–D	20.95
Flip 'n' File–Cart	20.95
Joysensor	Call
WICO Trakball	37.95
KRAFT Joystick	15.95

18 years of reliable service in:
Plainview, Syosset, New Hyde Park,
West Hempstead, Huntington, Patchogue

TO ORDER CALL TOLL FREE
IN NY STATE CALL TOLL FREE
1-800-221-2760 OR 516-249-2120

1-800-631-1003

Check, Money Order, MC or VISA accepted / No additional charge for MasterCard and Visa / Call for shipping and handling information / NYS residents add applicable sales tax / Prices and availability are subject to change without notice / All factory fresh merchandise with manufacturer's warranty / Prices & policy may vary in retail stores. Dealers Welcome.

Price reductions may have been made since this ad was placed.
Call us for our latest price quotes.

*Call for recent price reductions and new rebate information

VISA MasterCard

31 East Mall Plainview NY 11803

COMPUTER CENTERS OF AMERICA

disk insertion slot. Squeeze these once and the disk is clamped in place. Squeeze them a second time and the disk pops out.

Internally, the drive mechanism by Chinnon sports a JVC direct drive disk motor. As with the 1541, the track one stop is mechanical; however, the anticipated head chatter is much gentler. Overall operation of the Commander II drive is very quiet with disk surface noise the predominant factor.

The main circuit board, right beneath the drive mechanism, has sockets for all of the chips, making for easy replacement. As with the 1541, the device number can be changed by cutting one or two printed circuit traces located at the very rear of the circuit board. The brief manual made no mention of this.

The internal power supply is in a self-contained module at the back of the drive. We found an unlabeled, unidentified fuse hidden within this module. This fuse should have been in an external fuse holder mounted on the rear panel. Fuse changing with the existing arrangement would be beyond the capability of most users. The power supply itself was surprisingly compact compared to the other drives we had looked at.

Of the two drives we worked with, one ceased functioning after a brief interval. The second unit developed an internal rattle. This turned out to be an improperly glued stepper motor nameplate which had worked itself loose. This was promptly replaced before it could cause any damage.

Documentation supplied with the drive was extremely sparse. The brief manual was devoted exclusively to LOADing, SAVEing, and formatting disks. No mention was made of any of the more advanced disk operations, file handling, or commands. The included test disk contained only a single program for performing a fundamental operational check on the drive.

Overall, the Commander II seemed to be a competent piece of hardware which could use a good boost from improved quality control. □

THE INDUS GT— A DRIVE AND A QUARTER

Indus Systems Inc.
9304 Deering Avenue
Chatsworth, CA 91311
Phone: 800-54-INDUS
Price: $399

Inasmuch as you're all bound to be wondering about the cryptic significance of the title of this review, we'll get right to the point. Indus Systems has taken the ramdisk concept one step further in their implementation of a single disk drive for the Commodore family of home computers. They have included several useful disk utility programs in ROM as a pseudo drive 1. These programs can be LOADed as if they were on a conventional floppy disk by specifying the drive 1 parameter in the LOAD command. For example,

```
LOAD "1:FIO&DW",8,1
```

will boot the Indus fast input/output and DOS wedge utilities.

The drive 1 directory which is accessed by LOAD "$1",8, will produce the following display when LISTed:

```
1  "INDUS GT C64     " ID 2A
1  "FIO&DW"             PRG
1  "FIO"                PRG
2  "FAST I/O"           PRG
1  "DW"                 PRG
3  "DOS WEDGE"          PRG
1  "FC"                 PRG
5  "FAST COPY"          PRG
0  BLOCKS FREE.
```

These programs are permanently stored in the Indus drive. They cannot be changed, nor can any additional programs be SAVEd on drive 1. This, in our opinion, merits the designation of an extra one quarter drive for the Indus GT. Placing these three utilities, *Fast Input and Output*, the *DOS Wedge*, and *Fast Copy*, in ROM makes them instantly available every time the drive is powered up.

The *DOS Wedge* is similar to the one supplied by Commodore. The commercial at (@) symbol eliminates the need to OPEN and CLOSE the disk drive command channel (secondary address 15) when issuing drive commands. It also serves to read and display the DOS error report. As with the Commodore Wedge, the Indus Wedge lives at 52224 ($CC00) in the C-64.

SPEED OF OPERATION

The *Fast I/O* utility is said to speed up disk LOAD and SAVE operations. As we mentioned earlier, there is no way to build a single disk drive that is inherently faster than the 1541 without modifying the computer in some way (with the exception of internal operations—disk formatting, for example, a computer-independent operation, takes only 21 seconds with the Indus GT). The *Fast I/O* utility, when LOADed into the C-64, performs the modifications to the computer's operating system which will allow for faster data transfers. The *Fast I/O* program lives at 51200 ($C800) in the C-64.

In actual use, *Fast I/O* sped up a straight LOAD by over five times. It had no appreciable affect on SAVE times. The actual benefit you will derive from the *Fast I/O* utility will depend on the specific way in which you use your computer. Since the *Fast I/O* resides in the C-64 RAM, it is linked into the operating system by modifying the system vectors stored in low RAM. As a result it will be disconnected by virtually all autorun commercial software. The *Fast I/O* utility will only be of benefit for LOADing BASIC or binary files which do not modify the system memory usage in low RAM.

Note that the *DOS Wedge* and *Fast I/O* are provided both as separate programs and as a combined utility. This gives the user maximum flexibility in configuring the system to suit his own requirements.

Lastly, the *Fast Copy* utility performs a full disk backup in under five minutes with four swaps of the disk. The limitation to this utility is the inability to back up a disk with any errors on it. Encountering a read error results in the immediate interruption of the program. The moral of the story is not to wait for your disks to start going bad before making a backup copy.



We're sure you're looking forward to trying out the programs in this issue of *Ahoy!* But we're equally sure that you're *not* looking forward to typing them in. If you're an average typist, that should take you upwards of 25 hours. Not counting time spent correcting your typing errors, of course. How long that will take is anyone's guess. • How would you like someone to type the programs for you? At a price of, say—32¢ an hour? Wouldn't it be worth 32¢ an hour to free yourself up for more pleasant pursuits—like enjoying the rest of your *Ahoy!* magazine? • If you order the *Ahoy!* Program Disk or Cassette, you'll be getting that service for just under 32¢ an hour. Because for $7.95 (postage included)* we'll mail you all the programs in this issue, on a disk or cassette that's tested and ready to run with your C-64 or VIC 20. If you subscribe, you'll save even more: 12 monthly disks or cassettes for $79.95, 24 for $149.95 (postage included)*. Back issues are available as well. • You already know how to type. Why not use the hours you spend with your Commodore to learn something new?

*Canadian and foreign prices higher.

Ahoy! *uses only*
maxell
premium quality disks.

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)

	U.S.A.	CANADA	ELSEWHERE
☐ April '85 *Ahoy!* Disk ☐ April '85 *Ahoy!* Cassette	$ 7.95	$ 12.95	$ 13.95
☐ 12-Month Disk Subscription ☐ 12-Month Cassette Subscription	$ 79.95	$ 99.95	$124.95
☐ 24-Month Disk Subscription ☐ 24-Month Cassette Subscription	$149.95	$179.95	—

Make check or money order payable to **Double L Software.**

BACK ISSUES

Jan. '84 issue $9.95; all other back issues $8.95. Indicate your choices below, and specify disk or cassette at the bottom. In Canada add $2.00 per issue; outside U.S. and Canada add $4.00 per issue.

☐ Jan. '84	$9.95	☐ ______	$8.95
☐ ______	$8.95	☐ ______	$8.95
☐ ______	$8.95	☐ ______	$8.95
☐ ______	$8.95	☐ ______	$8.95

SPECIFY
☐ *DISK* ☐ *CASSETTE*

N.Y. State Residents must add 8¼% sales tax.

Send coupon or facsimile to:
***Ahoy!* Program Disk/Cassette**
Ion International Inc.
45 West 34th Street, Suite 407
New York, NY 10001

NAME ______________________
ADDRESS ______________________
CITY ______________ **STATE** ______ **ZIP** ______

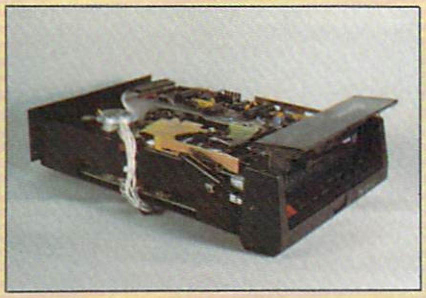
BELLS AND WHISTLES GALORE

The ads for the Indus GT make reference to a certain foreign automobile. We suppose this has something to do with all the "accessories" included with the GT that were left off the 1541. Or the sleek futuristic styling or the compact size of the drive. Its overall dimensions of eleven by seven by three inches result in a volume just about forty percent less than that of the 1541. A part of this reduction in volume is simply due to the fact that the power supply has been placed outside the drive. The power supply is very similar, in overall size and construction, to the battery eliminators you may purchase for your portable radio or cassette player. It plugs into a small connector on the back of the GT right next to the on/off switch. With the external placement of the power transformer, a major source of heat has been removed from within the confines of the drive. Nevertheless, the Indus GT did become cozily warm during normal operation. As with the C-64, there is no convenient way to turn the power supply off without pulling the plug or using an external power switch.

The drive itself has a black crinkle finish on an aluminum and plastic enclosure. It comes complete with a transparent hinged front door which opens in a very leisurely high tech fashion by the simple action of a built-in spring and piston. This door acts as a combination dust cover and sound damper. We found that the drive operation was extremely quiet even with the door open. Most noises were generated by the disk surface itself. The manual also touts the value of the door in preventing inadvertent access to the drive mechanism during critical disk operations.

We found the dust cover door to be quite durable, as it survived unscathed the attentions of this reviewer's two-year-old daughter, who derived great pleasure from operating the mechanism for about fifteen minutes. This was estimated to be equivalent to two years' normal operation for the dust cover hardware.

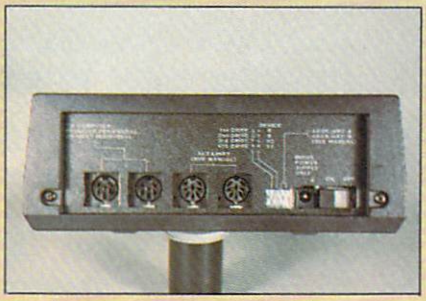
Under the door, right below the



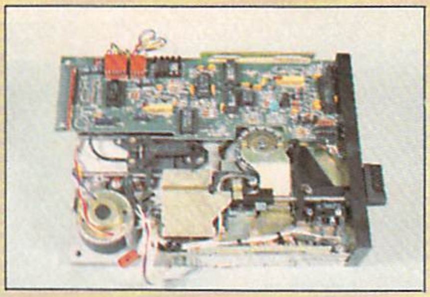
The Indus GT disk drive revealed.

mandatory orifice which accommodates the insertion and removal of the disk, is a control panel complete with buttons and blinking lights. These are not just to give the black drive better visibility in a darkened room. At the extreme left is a two digit light emitting diode (LED) display. This normally indicates which track the drive is currently working on. The track display works quite well, although it occasionally will not read properly with certain forms of copy protection. If a disk error should occur, the display flashes the pertinent DOS error number.

Right next to the display is a rather redundant power indicator light. This serves no useful purpose, as it is invariably overwhelmed by the visible radiation from the two digit display which always provides some vital bit of illuminating information when the power is on. Moving along to the right, a "busy" light is provided which indicates when the drive is active. The next indicator light is most useful. It comes on if the disk has a write protect tab covering the write enable notch. This light also works in conjunction with the first of four buttons to the right. This button, which toggles a hardware write protect function, allows you to write



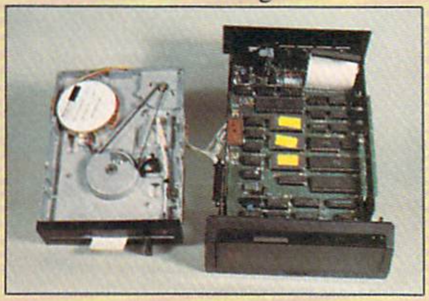
Back view, l to r: two serial port connectors, two mystery 8 pin din connectors, device no. setting switches, mini power jack, on/off switch.



Top view of the drive mechanism.

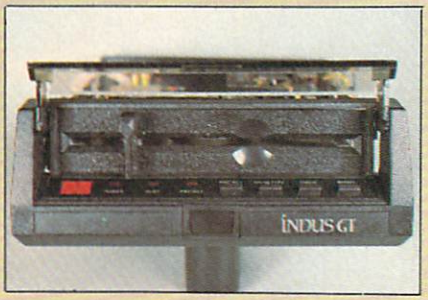
protect a disk without actually covering the small notch with a piece of opaque tape.

The three remaining buttons control the data which will be shown on the LED display. The "drive type" button causes the current device number to be displayed, normally an 8. The "track" button causes the current track to be displayed, the default condition. The "error" button displays the DOS error message number.



At left is the drive mechanism; at right is the main circuit board.

The hardware enhancements do not stop with the front panel. The back panel has a lot more going for it than the one on the 1541. A set of miniature switches is located right next to the power jack. Two of these are used to set the drive to any device number from eight to eleven. The remaining two are for an undefined



Control panel, l to r: lights—2 digit LED, power, busy, write protect; buttons—write protect, display device no., display track, display error.

auxiliary function (see below). At the opposite end are the mandatory dual serial port connectors for connection to the computer as well as for chaining to additional serial bus peripherals. In the center are two eight pin DIN connectors. These are simply labeled as "auxiliary (see manual)". Seeing the GT manual beside the computer, we did just that, and found little additional enlightenment other than that these have been reserved for future use. A brief speculative conversation with an Indus representative resulted in our being sworn to secrecy. However, since we expect that all our readers can count, we will merely point out that there are eight pins in those DINs and there are eight bits in a byte. Enough said!

BEYOND THE HARDWARE

The manual provided with the drive does a decent job of describing the basic setup. It also gives a rudimentary explanation of the BASIC DOS commands and their purpose. Complete instructions are provided on using the built-in software, as well as the GT Utility Diskette which contains a single multipurpose disk utility program. This program, writer by Mike Louder of Datamost, supports all the basic disk maintenance functions such as renaming, copying, deleting, and undeleting files. Both one and two drives are supported. Also included is a rather nice track and sector editor for more advanced users. This last utility displays the contents of a disk block, in both hexadecimal and character format, spread out over two screens. A hard copy of the sector contents can be made on a Commodore 1525 or any other interfaced ASCII printer.

What the manual lacks is any instruction on how to read and write disk files. This is mitigated somewhat by the inclusion of an additional book with the drive, *The Elementary Commodore 64* by William B. Sanders. This Datamost publication, which contains many interesting little tidbits, includes a brief chapter on the use of sequential files. Some recommendations for additional reading are included at the conclusion to these reviews (see page 82).

ADDITIONAL SOFTWARE

Packed with the Indus GT are a set of four disks. One of these is the GT Utility disk discussed above. The other three comprise a complete set of productivity software consisting of the *GT Estate Word Processor,* the *GT Data Manager,* and the *GT Albert E. Spreadsheet* for the Commodore 64. Each of the programs was supplied with its own manual. We do not have the space to go into a detailed review of each of these programs at this time. However, we will say that, overall, the programs are quite effective.

The word processor, usable only with the Commodore 64, includes many advanced features. Among these are movement of text blocks, embedded format commands for text output, merging of disk files into an existing document, and search and replace. During text entry, the word processor is always in an insert mode. Placing the cursor anywhere in the text causes all data to be inserted at that point. Deletion of text is accomplished either by a character, word or a 256 byte block. All deleted text is temporarily held in a 256 character buffer for recall at the same or a different location. Linked files are not supported. Most users will find the large, 39,000 character edit buffer to be more than adequate for their needs.

Word processor text is saved as a program file in screen display code format. Text file names are also in display code format. As a result, the directory listing of a word processor disk looks very peculiar. The files are also inaccessible to normal DOS commands based on keyboard entry. Interestingly enough, we discovered that the *GT Estate Word Processor* was incompatible with the Indus *Fast I/O* utility. Although the program seemed to boot normally, it was unable to properly recall saved text files.

The data manager is written in BASIC. It will work with both the VIC 20 with at least eight kilobytes of expansion and the Commodore 64. Nevertheless, it was effective and easy to use. It did allow a user to break out of the program with the risk of a loss of data. All data is maintained in memory, limiting the capacity of the program as compared to disk-based systems. This did have the advantage of speeding up data entry and item searches.

The spreadsheet program will also run on both the VIC 20 with at least sixteen kilobytes of expansion and the Commodore 64. A fully expanded VIC 20 has room for 35 rows by 50 columns and displays two columns of spreadsheet data. The Commodore 64 supports 35 rows by 99 columns and displays four columns of spreadsheet data. Ten mathematical operations are supported, including the four basic operators as well as percent, percent change, row or column totals, averages, and high and low values.

One thing which puzzled us was the need for placing the included software on four separate disks. None of the programs were excessively long. In fact, the contents of all four disks were not sufficient to fill half of a single disk. This turned out to be another subtle bonus from Indus Systems. The lack of any write protect tabs on any of the disks was the initial clue. The obvious intent was to give the user an opportunity to exercise the accompanying disk utilities by creating a backup disk containing all of the software. Since one of the programs included a mild form of copy protection, this also gives the user some incentive to learn some of the more subtle intricacies of the disk drive. We congratulate Indus on their thoughtfulness and foresight.

RELIABILITY AND COMPATIBILITY

These two issues are prominent in the minds of many existing 1541 owners in search of a second disk drive as well as those still contemplating the purchase of their first. Long-term durability is of course impossible to ascertain with a brief test of this sort. We will make several observations. A considerable amount of thought seems to have gone into



the design of the drive. The track one indicator is optical, rather than mechanical as in the 1541 drive. This should help preserve long term alignment of the read/write head. The drive mechanism is a belt driven model from Tandon, an experienced manufacturer of floppy disk drives. All of the chips were socketed for easy replacement. This does add somewhat to the initial cost of the drive and is most likely a temporary measure until full production is reached.

One interesting, although somewhat disconcerting, touch is the spinning of the disk spindle every time a disk is inserted or removed. The purpose of this is to insure proper centering of the disk on the drive spindle. This was a recommended practice with the earlier CBM disk drives, although we have never felt a need to continue it with the 1541 given the nearly universal application of disk hub rings.

We do feel obligated to mention that the first sample which we received was apparently defective. It had a definite sensitivity problem when reading disks, causing it to report good data as unreadable. The replacement unit has performed flawlessly.

Compatibility with the 1541 disk drive appears to be quite high. We were able to LOAD all the copy-protected software we had available at the time. This included new material from Electronic Arts, Datasoft, and Epyx. It even worked with such esoteric utilities as *Disector* and *Super Clone with Tough Nuts Utility*. This is of course no guarantee that future forms of copy protection, designed for the 1541, will continue to be compatible with the Indus GT.

CONCLUSIONS

The Indus GT is a lot more than just another single disk drive for the Commodore computers. The built-in software utilities and hardware features make it a useful addition to a growing Commodore system. The included productivity software will allow new users to get some immediate and rewarding returns for their investment. □

TWO HANDY DISK DRIVE ACCESSORIES

On occasion, one comes across a product or an idea which is unsurpassed for its elegance or simplicity. Very often, upon demonstration, the concept is so readily apparent that you cannot help but wonder why you did not think of it first. Here are two products which are apt to evoke just such a response from many users.

1541 PHYSICAL EXAM

Cardinal Software
13636 Jefferson Davis Hwy.
Woodbridge, VA 22191
Phone: 703-491-6502
Price: $39.95

Much has been said about the possible ills which afflict users of the Commodore 1541 disk drive. The most common bugaboo is misalignment of the read/write head, that electromagnetic gizmo responsible for getting information from and placing information on the floppy surface.

Misalignment in the 1541 occurs when the stepper motor pulley slips slightly on the stepper motor shaft. It is caused by the mechanical banging of the aforementioned pulley on what is known as the track 1 stop. The loud chattering noise you hear when formatting a disk is one result. This operation allows the disk drive to place the head at a known position. At certain times this operation is unavoidable, as when formatting a disk. The most common cause of the head chatter is the various forms of copy protection. The most common cause of misalignment is probably an overindulgence in attempting to copy such software. More recent forms of both copy protection and copy programs are far more gentle on the disk drive.

Misalignment can be recognized by an occasional flickering of the red light when LOADing long programs. As the condition worsens, the flickering increases till the point is reached where the disk becomes unreadable. At this time the disk drive reports some form of read error. The uninformed user is frequently misled into believing that the older disks are going bad. This is because the more recently formatted disks are still perfectly readable. This is where the greatest danger lies. Disks formatted on a misaligned drive can only be read by the same drive. Once the drive is repaired, the old disks are readable again, but the new ones are not.

Until recently for most users, the alternatives have been rather limited. Either send the drive out for repair, a time-consuming, costly proposition, or if you have the necessary electronics knowledge and test equipment, simply do the job yourself. For the former, Commodore is probably the best choice for a sick drive. The $85 replacement fee may be a bit steep for a simple alignment, but it does buy you a brand new or completely refurbished unit with all of the latest upgrades. For the latter, the *1541 Single Drive Floppy Disk Maintenance Manual* (see reference 3) is a veritable gold mine of information on the physical workings of the 1541.

Now, Cardinal Software's *1541 Physical Exam* reduces the alignment problem to the mechanical procedure it actually is. If you are reasonably adept at taking things apart, this package will let you align a 1541 without any test equipment whatsoever. All that is required is a good quality Phillips head screwdriver.

The package contains a specially formatted disk with a simple operating program. The disk also has tracks 1, 16, and 35 specially formatted. These tracks have been recorded with the odd and even numbered sectors offset by integral multiples of one thousandth of an inch. The program, by checking these offset blocks, is able to precisely measure the state of alignment of your drive.

The really nice thing about the package is that it lets you check the mechanical condition of the drive in less than five minutes—without any tools and without opening the drive. The test covers disk rotational speed and the track one stop position as well as head alignment. An easy to read, no-nonsense screen display de-

Continued on page 82

Increase the speed of your BASIC programs by 10 to 45 percent with

FASTER 64

By Anthony Wood

Faster 64 is a utility for the Commodore 64 that analyzes the use of variables in a BASIC program while it runs. It tells you how often each variable in the program is accessed. By defining your most often used variables first, BASIC does not have to search as far for them, thus speeding up your program. For some programs, this speed increase can be considerable.

Faster 64 is a machine language program. Since it is slow and inconvenient to use a BASIC program to POKE in a machine language program, we have created a special ML listing of *Faster 64.* To enter *Faster 64,* refer to the *Flankspeed* instructions on page 94. Once entered and SAVEd you need only enter 'LOAD "FASTER 64",8,1' (tape users, 'LOAD "FASTER 64",1,1') when you wish to load the program. To initialize it, you also have to type 'SYS 49152'. This should be done after you first LOAD it or after you press RUN STOP/RESTORE.

After you key in the enclosed program, *Faster 64,* you should save it before you run it. Once you have saved it, LOAD it and type SYS 49152, and you should see the message "*Faster 64* working."

USING *FASTER 64*

Load *Faster 64* and initialize it. When you see the message "*Faster 64* working," enter the following line:

```
Q=0:A=0:A(1)=A(2)+A(3):Z$="FRED"
<RETURN>
```

The following should appear:

```
A() 3 ,Q 1, A 1,Z$ 1
```

This means the array A was referenced 3 times and the variables Q, A, and Z$ were each referenced 1 time. The variables referenced the most are listed first.

Key in this short program to test *Faster 64* some more:

```
NEW
10 DIM A(20)
20 FOR Y=1 TO 20
30 A(Y)=A(Y)+1
40 NEXT
```

RUN it. You should get the message:

```
Y 41, A() 40
```

This means that the array A( ) was referenced 40 times, and the variable Y 41 times. Notice that a FOR-NEXT loop only references its index once. This is because the FOR-NEXT loop stores the address of its index variable. It does not have to keep looking it up. It will, however, look up the index in every loop if you enter 'NEXT Y' instead of just 'NEXT.'

You should be aware that the variable TI will not work with *Faster 64*; it causes a syntax error.

To use *Faster 64* on one of your programs, load *Faster 64* and initialize it. Note that you might have to enter "NEW" after you load *Faster 64,* to prevent the "out of memory" error. This is a bug in the Commodore BASIC ROM. Now load your program. Run your program all the way through. After your program is finished, its variables will be listed in numerical order. Suppose that you run your program, and you get the following display from *Faster 64:*

```
X 2131, Z 511, P() 200, F()154, X$ 100,
D 2
```

To initialize the variables in the correct order, you would enter a line at the beginning of your program like this:

```
1 DIM P(100),F(100)
2 X=0:Z=0:X$="":D=0
```

This puts your variables in the most efficient order. Notice that the arrays are on a separate line.

You should look out for certain exceptions. For example, suppose you find out that the variable A$ is referenced 4000 times. It might not be best to define it first, if it is not at a place in your program where speed is important. For example, suppose A$ appears in this line:

```
1000 GET A$:IFA$<>CHR$(13)THEN1000
```

You can see that A$ is in a loop waiting for a return. Since it is used in a wait loop, you can define A$ last because speed is not important—defining it first would just slow down the search for more critical variables. □

SEE PROGRAM LISTING ON PAGE 106

BOOK REVIEW

ASSEMBLY LANGUAGE FOR KIDS by William B. Sanders (Microcomscribe, 1984; $14.95 without disk, $24.95 with disk).

Trying to learn assembly language without attending college might seem an impossible dream, but don't believe it. Most assembly language programmers learn by trial and error and home study methods. Until a few years ago, though, the books available on the subject were aimed at technical people who already knew how a particular microprocessor worked and had a good idea of how to program it. Few good texts have been written for the layman; this is one.

Don't let the name fool you. This book is not aimed at seven year olds. I passed that age twenty years ago and still found the book instructive. The author assumes his reader is an intelligent and inquisitive individual who has a working knowledge of BASIC programming and wants to move on to something more challenging. Whether the reader is seven, twenty-seven, or fifty-seven is inconsequential because this is an excellently written book.

The author has done everything possible to make assembly language comprehensible to informed computer users. By including a "Kids' Assembler" written in BASIC, the author quickly gets the reader actively involved in programming.

Although the assembler and sample routines are available on disk (for an additional $10), a chapter is devoted to keying in the assembler and understanding how it works. The instructions are excellent, explaining what each section of the program does and the variables used, and relating each part to the whole. It's a very limited assembler, with no editing facilities or comment fields, but this keeps the reader's typing to a minimum. The step-by-step approach he uses is very effective, without being condescending.

The author has a great way with analogies and comparisons. His explanation of the hexadecimal, decimal, and binary number systems is one of the best I've seen. His sections on using an assembler and designing programs are also excellent. He clearly explains the commands and how to use them. Before it's through, the book will have you manipulating sprites and other screen graphics and using the joystick port for input.

The text is liberally sprinkled with sample programs encouraging you to really learn what you're reading. Cute illustrations with witty captions break up the text and keep things from getting too laborious. Programs in BASIC and assembler that accomplish the same objective let you compare their speed and size.

This is one of the best $15 courses in assembly language available. If you think you're a little too old to be called a kid, just grin and bear it. We're all kids at heart, right?

Microcomscribe, 8982 Stimson Court, San Diego, CA 92129. *—Cheryl Peterson*





PHOTOS BY ROBERT FERRONE



BOOTER
for the VIC & 64

By George Jones

Here's a chance to add a little machine language magic to your own programming efforts. You'll be able to load and run your programs with a single command. There are versions of the *Booter* for the C-64 and VIC 20, and you need no knowledge of machine language to use them. For those of you who are interested, an explanation of the technique is included.

Many commercial software packages include what is known in the trade as a boot. It is simply a short program that loads the main program or programs that make up the software package. The boot may also perform some of the setup procedures required by the main program and may even RUN it. The C-64 Wedge is a good example of a boot. It loads and runs the well-known DOS 5.1.

The *Booter* will write a fast machine language boot that will load and run the program you specify, provided you save it onto the same disk! Your chosen program may be BASIC or machine language. If it is machine language you must be able to supply the SYS address when the *Booter* asks for it. How does the program start without my typing RUN, you ask? We must play a little trick on the operating system of the computer to get it to do our bidding.

After the Kernal's power-up activities are completed, control of your machine is turned over to a BASIC ROM routine at $A483 ($C483 for the VIC 20) that will process the next line of BASIC you type. The address of this routine, $A483, is stored in operating system RAM at $0302-$0303, the BASIC warm start vector. By changing this vector you can send the computer to neverneverland or to the start of the boot written by the *Booter.* Since the boot is machine code, the explanation of its operation will seem a bit foreign to those of you who are uncomfortable with hexadecimal numbers and the like; but the *Booter* itself is very friendly and won't ask you any tough questions.

When the boot is loaded into operating system RAM, it alters the BASIC warm start vector. The new address sends the computer to the start of the routine that will LOAD and RUN the program we have named. The boot must also restore the BASIC warm start vector to its power-up value. This will allow normal operation of the computer when we're through with the program we are about to boot. We must call three Kernal routines to load the program of our choice into memory. They are SETLFS, SETNAM, and LOAD. The *Programmer's Reference Guide* will supply you with more information on these and many other user-callable Kernal routines.

If we have booted a machine language program, we must reset the pointer to the start of BASIC variable storage due to the action of the automatic relocator in the C-64 and VIC 20. Neglecting this step will cause an out of memory error at the next input. Finally a simple JMP to the entry point completes our task.

To boot a BASIC program requires some additional setup. We must also call two BASIC ROM routines. The code starting at $A659 ($C659 for the VIC 20) will reset BASIC execution to the start of the (BASIC) program. The next routine we call, at $A7AE ($C7AE for the VIC 20), is the BASIC interpreter loop, which will RUN the (BASIC) program our boot has loaded. Some of you may have noticed the difference between the addresses of the ROM routines is exactly $2000. Recall that the BASIC ROM of the C-64 begins at $A000 and that of the VIC 20 begins at $C000. A word of caution here: all translations of C-64 to VIC 20 BASIC ROM routines are not that straightforward. You will need to obtain some rather complete memory maps to properly translate some other ROM routines.

With the boot saved as the first program on the disk, next save the program you want to boot to LOAD and RUN. Now type LOAD":*",8,1 and press RETURN and watch your program LOAD and RUN. The ":*" tells the drive to load the first program on the disk in drive 0. If your drive is device 9, 10, or 11, then use the appropriate number. The boot will not prevent you from loading and running a program in the conventional manner later in your session if you so choose. If you would like to use several different boots for different programs on the same disk, and you feel like experimenting, you will simply replace the formatting command in the program listing with OPEN 15,dv,15,"I":CLOSE15. That requires changing the program name of the boot to something meaningful for your particular application. Instead of *The Magic Boot,* call it "BOOT PRG NAME,P,W". Because Commodore allows 16 characters maximum for a filename, your title shouldn't exceed 11 characters. This option requires a disk that has already been formatted and you will have to type LOAD"BOOT PRG NAME",8,1.

The *Booter* itself is all BASIC. It POKEs the boot into memory and customizes it according to the answers you give to the questions it asks. It then formats a disk and saves the boot. It should prove to be a great aid in cutting down keyboard errors for young users, poor typists like myself, and people who can't remember the correct spelling of program names or the proper SYS address. If you don't fall into any of those categories, you'll just have to settle for that little touch of pizzazz the *Booter* can add to your programming. □

SEE PROGRAM LISTINGS ON PAGE 103



HAS EVERYTHING!

SUBSCRIBE TO AHOY!

☐ Twelve Issues for $19.95 ($26.95 Canada and elsewhere)
☐ Twenty-four Issues for $37.95 ($49.95 Canada and elsewhere)

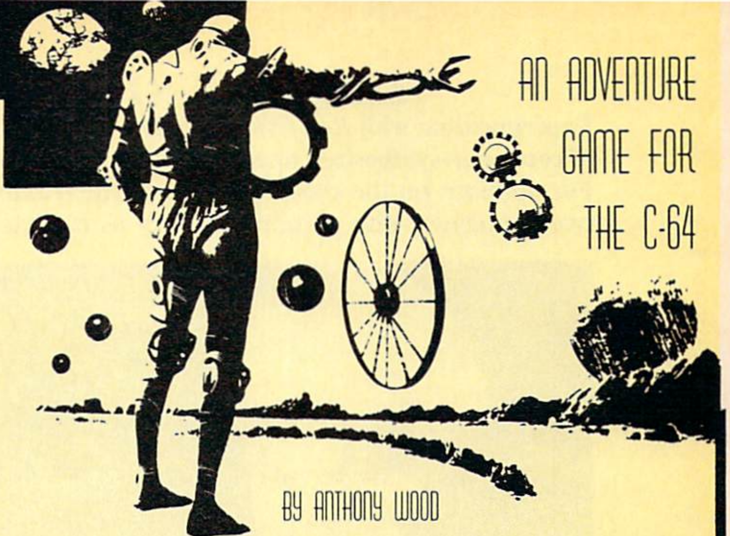
Name____________________

Address____________________

City__________ State__________ Zip__________

Send coupon or facsimile to:
ION INTERNATIONAL INC.
45 West 34th Street, Room 407, New York, NY 10001

Space Hunt



AN ADVENTURE GAME FOR THE C-64

BY ANTHONY WOOD

Adventures have been around for almost as long as the microcomputer. *Colossal Caves,* one of the first, was created on a mainframe, and has since been translated for almost all micros. A man named Scott Adams happened to play this original adventure. He loved it so much that he decided to write an adventure for his TRS-80 Model I. Thus emerged Adventure International and the Scott Adams adventure series. Scott Adams' first release, *Adventure Land,* was a hit and led to many more great adventures.

More recently, the so called "second generation" adventures have emerged. These games, such as *Zork, Starcross,* and *Witness,* allow the player to enter sentences of more than two words, like "Put the glass on the table." The first adventures were limited to two-word sentences.

These new adventures also contain more detailed information and more complex plots. They take a very long time to solve–weeks or months. The first adventure could be solved in five hours by a good player.

Although the new games offer better storylines and easier command structures, I still prefer the old style. Those adventures were more of a puzzle to me–half the problem was figuring out what words to use. They also could be solved in one or two sittings–a feature I like.

In the tradition of the first series of adventures, I have written *Space Hunt.* Your mission is to find the treasure while staying alive; admittedly, an old theme, but an entertaining one. *Space Hunt* is fairly easy and short–it is designed for the beginning to intermediate player.

For those of you that have never played an adventure like this before, I will explain how it works. The computer will describe your location and obvious exits at the top of the screen. You can now enter a direction to move in, or a two word command.

Directions can be entered as north, south, east, west, or n, s, e, w. These are the obvious exits–other directions might be available. For example, you could say GO DOOR to go to a door.

Some possible words are QUIT, SAVE, LOAD, LOOK, or EXAMINE. To save a game to disk, you can enter SAVE or SAVE <filename>. To load a saved game, enter LOAD or LOAD <filename>. The description of your location can be reprinted in two ways–press SHIFT CLR or type LOOK.

Some other common commands are GET item, DROP item, or INVENTORY (or I) to show what you are carrying.

The program looks only at the first three letters of a word, so you could abbreviate SHOOT ALIEN as SHO ALI. Also, when you type in the program, some lines will not fit in 80 columns. You should be able to enter these lines by using a "?" instead of PRINT.

Good luck with *Space Hunt*–I hope it gets you started on some adventures of your own. ☐

SEE PROGRAM LISTING ON PAGE 100

All the programs in this issue of *Ahoy!* are available on disk or cassette. See page 33 for details.

ELECHECK

Electronic Checkbook Management for the C-64

Each year as that black day in April looms near, a mad scramble takes place in the attempt to find receipts, checks, and other information related to tax deductions. This program will allow the user to catalog checks in twelve preset categories (easily changeable), have monthly graphics comparisons, print out the entire file or one record with totals, and as an added bonus, print the monthly graphics comparisons.

After the checks are entered, the user can enter the checkbook balancing routine and reconcile his checkbook balances. The information asked for in the program is check number, check date, check payee, check purpose, and check amount. The user can examine the check file or one record.

Elecheck uses sequential files to store the information on disk. I felt that since the size of these files would not be extremely large, there was no need for relative files, as sequential files are just as fast in this size range. A word of caution: the files must be initialized before use, and each program as the initialization before use, and each program has the initialization routine in each menu. The prompts make the program easy to follow, with each module changing colors to further inform the user that a different routine has been entered.

Elecheck can easily be tailored for monthly budget planning. ☐ ***SEE PROGRAM LISTING ON PAGE 108***

By Glenn Lumpkins

THE EXPERTS ON MUSIC PROGRAMMING

Experimenting with *Roll Over Pachelbel* (page 91 in this issue), just one of the many examples of the versatility of computer-synthesized music, got us to wondering just how much longer oboes and violins would be around. For opinions on the clash/synthesis of the traditional and modern methods of making music, we spoke to two gentlemen who certainly qualify as experts in both.



PETER NERO

Renowned pianist/conductor/composer Peter Nero is also the world's foremost celebrity electronics addict. An official spokesman for the electronics industry, he seemed the perfect authority to question on how electronic music compares to the traditionally produced variety in the eyes of a professional.

"Something is missing when there isn't direct human contact in producing sound," says Nero. "I have nothing against synthesizers, even though you can always tell the difference between synthesized drums, guitars, etc., and the real thing. It's not in the quality of the sound, but in the way it's made, that something's missing. A good analogy is the difference between a piano and an organ. An organ is a trigger mechanism. Even pressure-sensitive organ keyboards are triggering something else. The piano, though, is an extension of the arm; a wind instrument, an extension of the voice. That's something *I* miss when electronics take over."

Nero does not feel that the current generation, weaned on electronic music, will come to accept it to the total exclusion of music produced by humans. "You can't say that the people who appreciate live music are older people who'll die off," says Nero. "Something is happening between the musicians and the people in the audience. Of course, you can do marvelous things with synthesizers that you can't do with instruments. Look at today's records that feature enhanced sound—and no coughs. But the benchmark is, direct contact. There'll always be a place for an acoustic instrument, a musician, a symphony. It's too big a part of people's lives not to go on drawing an audience." □

RYO KAWASAKI

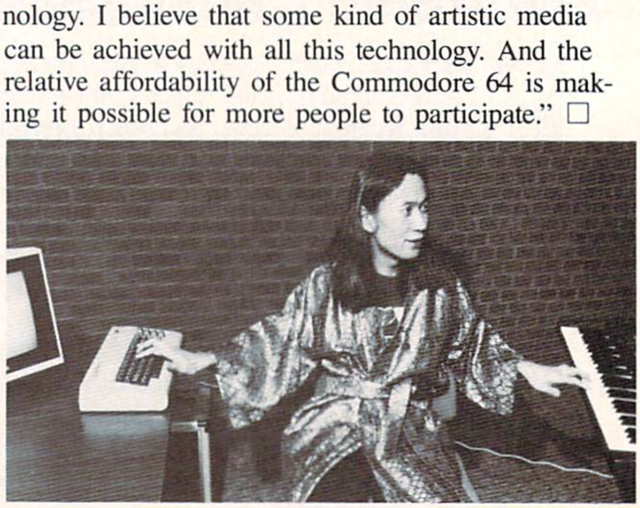
Well-known jazz guitarist Ryo Kawasaki began experimenting seriously with electronic music in 1978. His 1980 album, *The Golden Dragon Live,* grew out of his work with the Roland guitar synthesizer. His search for new sound possibilities led him to the C-64, and within a short time he began designing the programs now being marketed by Sight & Sound Music Software, Inc.

Kawasaki had done no programming prior to acquiring his C-64 in October 1982. "I am basically a musician," he explains. "When I began dabbling with it in the beginning I was not a very good programmer at all. I was working with BASIC and didn't even know about machine language."

Working with machine language has allowed the meticulous Mr. Kawasaki to get on truly intimate terms with the Commodore 64. His attention to detail is astounding and his dedication is inspiring. In the heat of creating a new program he can spend up to 15 hours a day playing with Op-Codes and orchestrating time. "I'm working in terms of two microseconds," he says. "That's how I relate to the machine. If any part of the program, any instruction, takes more than 16 milliseconds, it's going to crash. So everything must be done within that crucial time."

Of his burgeoning interest in software programming, Kawasaki says: "The problem all along has been that programmers have the technique but they don't know the music. And while musicians know the music, they don't have the same adeptness with the technology. So for all this time there has been no communication. I often thought that if a great musician and a great programmer could get together they could produce something outstanding. Now I'm communicating with both worlds.

"Unlike some musicians, I am not afraid of technology. I believe that some kind of artistic media can be achieved with all this technology. And the relative affordability of the Commodore 64 is making it possible for more people to participate." □



Getting into the Kernal

Accessing Machine Language Routines from BASIC

By Dale Rupert

Are you ready for an intellectually stimulating journey into a world beyond BASIC? The world of assembly language programming is a place where the instructions are three-letter mnemonics such as LDX and JSR. It's a world dealing with such strange sounding creatures as stack pointers and psuedo-ops. Programs written in assembly language allow the computer to perform at its absolute maximum speed. Programs containing one improperly placed bit can just as quickly send the computer off into never-never-land, requiring that the programmer power down to regain control of the machine.

This month we will first see how to access from BASIC some of the routines, called the Kernal routines, which are already built into the computer. We will discuss some of the concepts needed to create an assembly language program. In the second part of this article next month, we will write an assembly language program to perform some of the functions of this month's BASIC program. You'll see that pseudo-ops and LDX are really not so strange after all.

Let's begin by writing a BASIC program which will save a portion of memory onto a disk file. In particular we will copy the Commodore 64's DOS Support Program (commonly called the Wedge) from its location in RAM onto a disk program file. Our program will not be limited to copying only the Wedge, however. By specifying starting and ending addresses, we will be able to transfer any portion of memory, including screen memory, to a disk file.

Using BASIC alone would be a burdensome task. Over one thousand PEEKs and PRINT#'s to transfer a single 1K chunk of memory into a sequential file would be easy enough to program, but built into the computer is a better way to save memory to a file. It is the Kernal routine called SAVE.

THE KERNAL

The Kernal routines are utilities used by the computer's operating system and are stored in ROM. These routines may be accessed from BASIC. After specified initial conditions are fulfilled, a Kernal routine is executed from within a BASIC program by using the SYS statement.

Here are the details. Beginning on page 293 of the *Commodore 64 Programmer's Reference Guide (PRG)* is a description of the SAVE routine and an example of how to use it. SAVE is one of the more complicated routines, since we must use two other Kernal routines, SETLFS and SETNAM, in preparation for it. To begin, we must first find out about SETLFS and SETNAM.

Whenever you execute a PRINT#5 command within BASIC, you are referencing logical file number 5. When you OPEN such a file, you also specify a device number (printer=4, disk=8, etc.) and, in some cases, a secondary address or command. The SETLFS routine performs the same functions. It sets up the local file number, the device number, and the command.

The OPEN command in BASIC also specifies the filename for the file to be written. SETNAM is the Kernal routine which handles that function. When the BASIC interpreter executes a statement such as

```
OPEN 8,8,8,"DOS 5.1"
```

it calls upon the SETLFS and the SETNAM routines to establish the channel to the disk program file called "DOS 5.1". To gain experience using the Kernal routines, we will use SETLFS and SETNAM explicitly rather than using the OPEN statement. This is strictly an academic exercise since using the OPEN statement would certainly be easier.

We want to create a logical file number 8 to the serial bus disk drive (device 8). We will simply be sending data to the file, so we do not need to send a command. From the instructions for how to use SETLFS on page 297 of the *PRG,* we must:

1. Load the accumulator with the logical file number.
2. Load the .X register with the device number.
3. Load the .Y register with the command.

Now we are getting into the realm of assembly language programming. What are these things called the accumu-

PHOENIX RED
SOFTWARE



ADVENTURE GAME TUTORIALS

Disk I

TEXT ADVENTURES

- ZORK I
- ZORK II
- ZORK III
- ENCHANTER
- SORCEROR
- STARCROSS
- SUSPENDED
- PLANETFALL
- DEADLINE
- WITNESS

Only $9.99!

Disk II

MEGAVENTURES

- ULTIMA II
- ULTIMA III
- ULTIMA IV
- RETURN OF HERACLES
- QUESTRON

Only $9.99!

Disk III

GRAPHIC ADVENTURES

- GOTHMOGS LAIR
- MASK OF THE SUN
- DEATH IN THE CARIBBEAN
- BLADE OF BLACKPOOLE
- CRITICAL MASS
- GRUDS IN SPACE
- ULYSSES AND THE GOLDEN FLEECE
- WIZARD AND THE PRINCESS
- ASYLUM
- THE INSTITUTE

Only $9.99!

Have you ever played an Adventure Game where you just plain got stuck and couldn't get any further? Well, here's the answer — Phoenix Red's Adventure Game Tutorials. They can be used as hints or they will tell you exactly what to do to get out of all those sticky situations. These Tutorials are written for **your** computer (A Commodore 64) not someone elses, so they **work.**

We pay postage and handling.
A must-have for all adventures.
Order yours today!

"We put out money where our software is, not our copy protection."

— PHOENIX RED

Send check or Money Order to:

Phoenix Red Software
1620 Bay Area Blvd. Suite 703
Houston, Texas 77058

Call 713-280-8113 for more info.

Reader Service No. 114

lator and the .X and .Y (or simply the A, X, and Y) registers?

The accumulator and the other registers are memory elements within the microprocessor itself. They are similar in concept to the memory register in a typical four-function calculator. Values may be temporarily stored in the registers and recalled when needed. The accumulator is a more general purpose storage register. The microprocessor can perform various mathematical manipulations on the data in the accumulator. The X and Y registers are often used for storing addressing data, although the SETLFS routine uses all three of these devices as simple data storage registers.

How can we put data into these registers? When BASIC executes a SYS command, it automatically transfers the data from three specific RAM locations into the A (accumulator), X, and Y registers. These locations are listed starting on page 318 of the *PRG.* The RAM addresses are 780, 781, and 782 for the A, X and Y registers respectively. Consequently, to put an 8 into the accumulator, first POKE 780,8 and then execute a SYS command.

The SETLFS instructions indicate that if no command is to be sent, the Y register may be loaded with the value of 255. Consequently we want to put 8's into the A and X registers and put 255 into the Y register before calling the SETLFS routine. If you study Listing 1 (see page 96), you will see that the RAM storage areas for the A, X, and Y register data are defined in line 80. Line 220 defines the data to be stored in those locations. Lines 230 and 240 POKE the data into RAM. Now we have met the prerequisites for using the SETLFS routine. Using the SYS command to jump to SETLFS will properly load our values from RAM into the three microprocessor registers, and then execute the SETLFS routine.

The SYS command includes an argument specifying the starting address of the routine to be called. The Kernal routines are similar in nature to BASIC subroutines. A BASIC program may include GOSUB 300 to jump to a subroutine beginning at line number 300. At the end of the subroutine, the RETURN statement brings the program back to the statement following the GOSUB. Similarly, SYS 65466 sends the program to the machine language routine starting in memory location 65466. The routine will end with an RTS statement (ReTurn from Subroutine) to transfer control back to the BASIC statement following the SYS statement.

How do we know the proper memory location to use with the SYS statement? The Call Address for each Kernal routine is listed in its description. That is the address the SYS statement references. The Call Address for SETLFS (65466) is defined in line 210 of Listing 1, and line 250 transfers control to that address.

To be technically accurate, we should note that the Kernal routines are stored in ROM. The Call Address for each routine points to a separate portion of ROM called a "jump table." The jump table contains three-byte JMP (jump) instructions that immediately transfer control to the actual start of each Kernal routine. Our SYS 65466 causes a jump to address 65466. The instruction at address 65466 is a jump to the SETLFS routine elsewhere in ROM. To investigate this, type

```
A = 65466 : PRINT PEEK(A),PEEK(A+1),PEEK
(A+2)
```

The results are 76, 0, and 254. In hexadecimal, this becomes 4C 00 FE, which the 6510 microprocessor interprets to mean JMP FE00. In decimal English this becomes "jump to address 65024."

At the end of the SETLFS routine is an RTS instruction (ReTurn from Subroutine). Execution branches back to BASIC and line 260, which prints an appropriate message.

The SETNAM routine allows us to specify the filename of the program file we will create with the SAVE command. We have one more complication when using the SETNAM routine. From its description on page 299 of the *PRG,* we must first:

1. Load the accumulator with the length of the filename.
2. Load the X register with the low order address of the filename.
3. Load the Y register with the high order address of the filename.

Certainly step 1 is no problem. Listing 1 defines the filename in line 140, and line 320 determines its length. The length of the filename is stored into the A register RAM storage area in line 360. But what are steps 2 and 3 talking about?

ADDRESSING COMPLICATIONS

SETNAM requires that the filename be stored somewhere in RAM, and that we tell it the beginning address of that RAM storage location. First let's figure out where we might store the filename. Then we will determine how to specify that address to SETNAM.

The two RAM storage areas most frequently used by C-64 programs are the cassette buffer (addresses 828 to 1023) and an otherwise unused RAM area above the BASIC ROM (addresses 49152 to 53247). Either place would be fine for storing the filename. Line 310 in Listing 1 establishes the starting address of our storage area, called a buffer, at address 49155. The choice is somewhat arbitrary.

Lines 330 and 350 put the filename "DOS 5.1" letter by letter into the buffer. The "D" is at address 49155, the "O" is at address 49156, and the final "1" is at address 49161.

Now the task is to provide the starting address of this buffer to the SETNAM routine. The problem is that one byte of data can store numbers only as large as 255, and we are dealing with the number 49155. The solution is that we must use two bytes to handle this number.

By now you probably know that hexadecimal (hex) numbers are frequently easier than decimal values when dealing with computer data and addresses (except in BASIC!). The hex representation of 49155 is C003, or

as it is often written, $C003. The "$" merely indicates that this is a dexadecimal number. The letters 'A" through "F" are used as digits corresponding to the values 10 through 15, since the hexadecimal numbering system uses digit values from 0 through 15.

We won't go into a discourse on hex to decimal conversions here. If these concepts are unfamiliar to you, it won't matter since we will let the computer do the hard work for us anyway. Basically what we must do is to break the number $C003 into two parts, the most significant byte (MSB) and the least significant byte (LSB). Since the number is in hex, that's very easy to do. The MSB is $C0 and the LSB is $03. If you convert those numbers back to decimal, the results are 192 and 3 respectively.

To make things even simpler, or at least more general, two functions are defined in lines 60 and 70 of Listing 1 which calculate the MSB and the LSB of any decimal value up to 65535. FNH(X) finds the MSB or "high order byte" and FNL(X) finds the LSB or "low order byte" for the value X. Look at the results when we divide 49155 by 256:

```
49155/256 = 192 REMAINDER 3
```

The integral part of the quotient has the value of the MSB we are looking for, and the remainder equals the LSB. Thus the definition of FNH(X) should be obvious:

```
DEF FNH(X) = INT(X/256)
```

If we multiply FNH(X) by 256 and subtract the product from the original value, we are left with the integer remainder. Consequently:

```
DEF FNL(X) = X - 256*FNH(X)
```

Now we can complete the requirements for calling the SETNAM routine. We must put the "low order byte" and the "high order byte" of the filename buffer's address into the X and Y registers, respectively. Lines 370 and 380 of Listing 1 perform these functions in the wink of an eye.

So far we have opened an I/O channel to the disk and have specified the name of the file that will be sent to the disk. Once again, using SETLFS and SETNAM was the hard way to do that. Lines 180 through 400 of Listing 1 are primarily meant to show how these routines are used. Rather than type them in, you may use these two statements instead:

```
200 OPEN 8,8,8,"DOS 5.1"
520 CLOSE 8
```

These statements replace the corresponding lines in Listing 1.

READY TO SAVE

We have completed the preparatory routines for the SAVE routine. The remaining prerequisites as listed on page 294 of the *PRG* are:

1. Load two consecutive locations on page 0 with a pointer to the start of memory to be saved (in standard low byte first, high byte next format).
2. Load the accumulator with the single byte page zero offset to the pointer.
3. Load the X and Y registers with the low byte and high byte respectively of the address of the end of memory to be saved. (Actually this value must be one greater than the highest memory location which is to be saved.)
4. Call the SAVE routine.

Once again, these instructions presume the programmer is familiar with assembly language concepts.

Hopefully steps 3 and 4 are understandable following our previous discussions with the SETLFS and SETNAM routines. One more than the address of the end of memory which we are saving (EADDR) is defined in line 160 of Listing 1. Once again we use the FNL and FNH functions to convert that address into its low order byte and its high order byte. The values are put into the X and Y register storage areas in lines 490 and 500.

Steps 1 and 2 require that we learn some more about the 6510 architecture. The basic problem is that we must specify two addresses, starting and ending, to the SAVE routine. We have already seen that each address takes up two bytes. Thus we have four bytes to identify and only three registers (A, X, and Y) to work with. The solution is to store the starting address in a certain area of RAM. We will then load A with "information" so the SAVE routine can find the locations where we stored the starting address. This "information" consists of a one byte "page zero offset."

A "page" of memory consists of 256 bytes. Memory locations 0 through 255 ($0000 - $00FF) are called "page zero", locations 256 through 511 ($0100 - $01FF) are "page one", and so forth. The "page zero offset" is simply a number between 0 and 255 which corresponds to a memory location on page zero. A page zero offset of 8 refers to the absolute address 8 ($0008). Similarly a page one offset of 8 refers to address 264 ($0108).

Our instructions are to load two consecutive page zero locations with a pointer. The pointer is to be the starting address of memory to be saved. On page 316, the *PRG* identifies locations 251 through 254 as "Free 0-Page Space for User Programs." Therefore we will use locations 251 and 252 to store the pointer to our starting address. As usual, the pointer consists of two bytes in low order/high order sequence.

Line 450 defines ZPTR as the zero page address pointer and gives it the value 251. Lines 460 and 470 POKE the low order then the high order bytes of the starting address into 251 and 252.

Step 2 of the instructions is to load the accumulator with the page zero offset at which we stored our pointer. Line 480 puts 251 into the A register storage area.

Now the X and Y registers are properly loaded with a pointer to one location past the end of memory to be

saved. A pointer to the start of memory to be saved is stored in two bytes on page zero. The A register contains the offset into page zero at which the starting pointer is stored. And we are finally ready to call the SAVE routine in line 510.

Assuming you have already loaded the Wedge into memory before running this program, the SAVE routine will copy the portion of memory containing the Wedge onto whichever disk is currently in Drive 0. The machine language Wedge will be saved as a program file with the filename "DOS 5.1".

We have covered some fairly difficult concepts here. As with any new language, time, study, and practice are required to become fluent or even comfortable. Hopefully, seeing these ideas discussed from a BASIC perspective with which you are already familiar will make the learning process easier.

Of what value is this SAVE routine? You should be able to put the Wedge or any other "normal" machine language routines onto any of your disks. Also be sure to put the Wedge loader onto any disk that you want to boot up from. (That is the BASIC program which loads and initializes the machine language portion of the Wedge, called *C-64 Wedge.*)

Another obvious use for this program is to easily save screen images on disk. Use a starting address of 1024 in line 150 and an ending address of 2024 in line 160. Give the file a name such as "SCRN1". You may also want to save the corresponding color memory using addresses 55296 and 56296 respectively and a filename such as "COLOR1".

To restore the screen image once it has been saved, simply execute LOAD "SCRN1",8,1 (or type %SCRN1 with the Wedge installed). You will probably want to first load the color memory similarly, if you saved it. You might instead type POKE 53281,C and PRINT CHR$(147), where C is a color value between 0 and 15, before or after loading your screen image. The POKE statement will cause all characters to be displayed with the color value of C. This example should get you started:

```
10 IF A=1 THEN POKE 53281,1:END
20 POKE 53281,0:PRINT CHR$(147);
30 A=1 : LOAD"SCRN1",8,1
```

The screen image will be loaded "invisibly." It will instantly appear when the POKE in line 10 is executed. Notice that the LOAD statement with the ",1" option in line 30 causes program execution to return to the start of the program. The first time line 10 is executed, A equals 0 and the program jumps to line 20.

We have discussed a number of assembly language concepts. Next month we will go through the process of editing and assembling a routine which is comparable to the BASIC SAVE routine we have just completed. From the examples this month, you should now be able to access most of the other Kernal routines from within your BASIC program. ☐

SEE PROGRAM LISTING ON PAGE 96

S.O.S.

Is there any way to suppress the division by zero error? When my 64 encounters a zero in division, the error message appears and the program stops. Please hurry...the boss is waiting.
—Dennis Duncan
Waukegan, IL

Dennis,

The division by zero error is not implemented in the Commodore's BASIC to frustrate users like yourself. There is a good reason for a division by zero error, for mathematically one cannot divide by zero; the answer is undefined. One way to get around this problem is to put an IF-THEN statement along with every division statement in order to check for a division by zero before it causes a fatal error. If a division by zero is detected your program can then branch to a subroutine to handle the condition.

I am going quite insane, and have no one to turn to, save *Ahoy!* The reason for this insanity is that I can't get any answers to these three questions:

1) Is the 64's SID chip the same as, or similar to, the sound chip in the ColecoVision? I have noticed many similarities.

2) Do these fast load programs actually speed up the 1541 (I hope)? Or, in other words, will, say *Fahrenheit 451* play any faster using *Kwik-Load?*

3) The 64 has 16 colors, yet games such as *Death in the Caribbean* seem to display more than 16 colors. Is it that only 16 colors are available at one time when programming? I have put off buying a graphics program until I get an answer to this one.
—Jim Cirile
New Port Richey, FL

Jim,

1) The 64's SID chip (the magic slice of silicon that makes all the music) was designed by the engineers at Commodore specifically for the 64. I doubt very much that it has any relation to the sound device found in the ColecoVision. Any similarities you may have noticed may stem from the fact that they both use similar synthesis techniques, resulting in a similar type of sound. 2) Programs that speed up the 1541 do really work. They function very well for user-written programs. As far as "canned" software goes, things get a bit tricky. What these "quick-loaders" do is rewrite portions of the disk operating system contained in the drive. If the copy-protection scheme that the software package uses rewrites the DOS (disk operating system) as well (which most do), chances are that the loader will have no effect on the loading speed of the software package. 3) There are only sixteen colors on the 64. Any additional colors you think you see probably result from a mixture of the 16 colors on the machine. By creating different patterns with the sixteen colors, many effects can be achieved.

Is it possible to hook a typewriter to my C-64?
—Todd Hutchinson
Carbondale, IL

Todd,

Years ago I remember devices that fit over a typewriter's keyboard and actually "pressed" the keys corresponding to signals from your computer. This was when letter quality printers were well over $1000 and beyond the reach of the hobbyist. Today, letter quality printers are available for a few hundred dollars, and daisy wheel typewriters with computer interfaces are available for a few dollars more. As a matter of fact, I recall a recent ad for a daisy wheel typewriter-printer for under $300. With prices like these for daisy-wheel printers and typewriters, I can see little reason to try to convert a typewriter to a printer.

By David Barron

COMMODORE ROOTS

A COMMODORE ASSEMBLY LANGUAGE COLUMN WRITTEN EXCLUSIVELY FOR *AHOY!* MAGAZINE

INSTALLMENT ONE: BREAKING THE LANGUAGE BARRIER

If you've done much programming in BASIC, you've probably wished more than once that you knew how to program in assembly language.

Well, if you've ever had that urge, you came to the right column.

By the time you finish the installment you're reading now, you'll be programming in machine language. And within a few months, if you keep reading *Commodore Roots,* you'll be well on your way to becoming a proficient assembly language programmer.

If you're intrigued by this prospect, you're probably eager to start learning assembly language.

So let's start learning assembly language–right now!

COMPUTER LINGUISTICS

Programming languages can be divided into three major categories: high-level languages, machine language, and assembly language.

There are many, many *high-level languages:* BASIC, Pascal, COBOL, Forth, Fortran, C and dozens–perhaps hundreds–more. High-level languages are not so named because they are particularly esoteric or profound. They're called high-level languages because they're several levels removed from machine language–the only language that a computer can really understand.

Machine language is made up of numbers–nothing but numbers. So it's not an ideal programming language. But it's your computer's native tongue.

Assembly language, as you'll soon see, is very closely related to machine language. In fact, assembly language is not really a full-fledged programming language at all; it's nothing more than a notation system that was designed to make machine language programs a little easier for humans to write and understand.

Now we'll take a closer look at each of these language families.

HIGH-LEVEL LANGUAGES

All popular high-level languages have one feature in common: they all bear at least a passing resemblance to English. BASIC, for example, is made up almost completely of instructions derived from English words: PRINT, LIST, LOAD, SAVE, GOTO, RETURN, and so on. Most other high-level languages also have instruction sets based largely upon plain-language words and phrases.

But computers can't understand a word of English; the only language they can understand is machine language. So, when you write a program in BASIC–or in any other high-level language–it has to be translated into machine language before a computer can understand it. This fact is not always obvious to a computer user, but it's true. You can program your Commodore in BASIC, all right, but only because your computer has a built-in interpreter that translates every line of BASIC that you write into machine language. If you didn't have access to this interpreter–or to some other utility for translating a high-level language into machine language–you wouldn't be able to program your computer in BASIC, or in any other high-level language.

You don't need an interpreter to write programs in assembly language. But you do need another kind of software package called an assembler. I'll say more about assemblers later on in this column.

BY MARK ANDREWS

Mark Andrews is the former electronics columnist for the *New York Daily News* and a veteran reporter, editor, and technical writer in the fields of personal computers and consumer electronics. His articles have appeared in hundreds of newspapers and dozens of magazines; he has also written 12 books, the most recent titled *Commodore Roots: Assembly Language Programming for the Commodore 64,* to be published in the first quarter of 1985 by Howard W. Sams & Co., Inc. Many of the topics covered in this new book will also appear in this series of columns.

MACHINE LANGUAGE

Machine language, as I've mentioned, is made up of nothing but numbers. In its purest form, in fact, machine language is composed of *binary numbers*—numbers written as strings of ones and zeroes. Here's a listing that shows what a machine-language program looks like when it's written in binary numbers. This program, for reasons that you'll discover very shortly, is called *HI.TEST.* In its binary code version, I call it *HI.TEST.BIN.*

HI.TEST.BIN
(HI.TEST Program, Binary-Code Version)

```
1 0 1 0 1 0 0 1
0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 0
1 1 0 1 0 0 1 0
1 1 1 1 1 1 1 1
1 0 1 0 1 0 0 1
0 1 0 0 1 0 0 1
0 0 1 0 0 0 0 0
1 1 0 1 0 0 1 0
1 1 1 1 1 1 1 1
0 1 1 0 0 0 0 0
```

THE HEXADECIMAL SYSTEM

As you can tell from the short listing above, when you've seen one binary number, you've seen them all. Binary numbers look so much alike that if you spent much time staring at them, you wouldn't be able to tell one from another after awhile. So machine code listings are rarely written in binary numbers. Instead, they're usually written in a closely related notation system called hexadecimal.

Hexadecimal numbers aren't based on the value of 10, as ordinary decimal numbers are. Instead, they're based on the value 16. In hexadecimal notation, the arabic numbers 0 through 9 are used in exactly the same way that they're used in decimal notation. In addition, however, the letters A through F are used to represent the values 11 through 16. You'll learn more about the hexadecimal system—and why it's used in assembly language programs—in later columns in this series. So there's no need to go into any more detail about hexadecimal numbers right now. But, just so you'll know what they look like, here's what the *HI.TEST* program would look like if it were written in hexadecimal numbers:

HI.TEST.HEX
(HI.TEST Program, Hex-Code Version)

```
A9  48
20  D2  FF
A9  49
20  D2  FF
60
```

WE'RE NOT FINISHED YET

The hex numbers in the above version of the *HI.TEST* program have exactly the same values as the binary numbers that were used in the binary version of the program. You may not have any idea yet what the *HI.TEST* program means, but you can now see quite clearly that the hexadecimal version of the program is at least a little easier to read than the version that was written in binary numbers.

Now that we've converted the *HI.TEST* into hexadecimal numbers, only one more step is needed to translate it into assembly language.

ASSEMBLY LANGUAGE

As I've mentioned a couple of times now, assembly language is very closely related to machine language. But the relationship between assembly language and machine language is not always obvious at first glance. Assembly language is not made up solely of numbers, as machine language is. Instead, it is written using three-letter instructions called *mnemonics.* So, to the casual observer, assembly language doesn't look a thing like machine language.

But looks can be deceiving; and in this case, they certainly are. For every three-letter instruction that is used in assembly language, there is a numeric instruction that means exactly the same thing in assembly language. In other words, there is a precise one-to-one correlation between the mnemonics that are used in assembly language and the numeric instructions that are used in machine language.

Because of the close relationship between machine language and assembly language, it is not difficult to convert a machine language program into assembly language or to convert an assembly language program into machine language. To translate a program from either of these languages to the other, all you have to do is change each assembly language instruction into a machine language instruction that means the same thing—or vice versa.

Here are two more listings of the *HI.TEST* program that illustrate the close relationship between machine language and assembly language:

HI.TEST.ASM
(Object Code and Source Code Compared)

Line No.	Object Code			Source Code	
1	A9	48		LDA	#72
2	20	D2	FF	JSR	$FFD2
3	A9	49		LDA	#73
4	20	D2	FF	JSR	$FFD2
5	60			RTS	

Look carefully at this pair of listings, and you'll see some close similarities. For reasons that will become clear in later columns, the letters and numbers in the two listings are arranged slightly differently. But a close examination will reveal certain patterns in both listings that are the same. Look at the object-code listing, for

An open letter to the readers of Ahoy Magazine Vincent Kurek President: The Ennon Corporation

My purpose in writing is to ask you to join me in shaping the future of the new and most unusual field in computer technology today: Artificial Intelligence.

This incredible power and spectacular creative potential are available to you, for your computer right now. However, there is an alarming possibility that such amazing technology which you have every right to, may not be available to you other that through this offer.

This is unfortunate but somewhat understandable due to the way technology is created. You see, only the business oriented corporation can finance research. It therefore is in a position to dictate immediate research goals. These goals are increasing profits through more efficient production. While valid, they are merely creative and do absolutely nothing to foster exploration in new applications. The result: technology is never used to its fullest potential. But what's worst of all is that these competitve corporations have absolutely no desire to share technology with each other, let alone with you. So, they don't. As a result, the infinitesimal amount of technology that finally trickles down to you is:

A. So expensive you are prohibited from procurring it
B. Shamefully inferior to the real thing

remember...you can buy high-tech consumer goods, but never the technology that creates it.

This same situation confronts you in the new Artificial Intelligence field, but with a difference: There is no true Artificial Intelligence for the home computer user! The few programs claiming to be Artificial Intelligence are really simulators. They are not the real thing. Possessing a mere token of the power and versatility, simulators are clearly not worth their expensive price.

I have tried repeatedly to convince my colleagues that it is in their best interest to release genuine Artificial Intelligence to the general public. The refinement, modification and adaptation as individuals create new applications would improve Artificial Intelligence tremendously. This would benefit everyone in the long run.

I have met with little success. Apparently, it seems that immediate corporate profit is more important than sharing technology with the public. Therefore, the Ennon Corporation stands alone in offering superior Artificial Intelligence programming directly to the home computer enthusiast.

Announcing AN-83: The "Thinking" Program

Believe me when I say AN-83 is the real thing. It is a true "thinking" program that receives an initial "knowledge base" from a data file read when AN-83 is started. Using inductive and deductive logical analysis, this amazing program deduces everything from that data and adds it to its memory. Conversing with you, AN-83 adds and combines with facts already known. It generates new conclusions not explicitly contained in its original knowledge base—just like your own thinking process! The result: it knows considerably more than the specific facts given to it.

AN-83 can also think about anything. It is virtually unlimited in its application. Think of your possibilities. The potential is limitless. In the right hands, AN-83 would revolutionize the adventure, strategy and other smart game-playing programs to say nothing of classic arcade games. On the other hand, AN-83 could be one of the most powerful business analysts available to the home computer.

FREE SOFTWARE

In addition, you will be receiving free, Eliza—the most amazing conversational A.I. program to date. Run this for your friends and jaws will drop with amazement. Eliza's responses are so human, it's uncanny. An entertaining program, Eliza will answer once and for all the question: What can your computer do?

How to Learn Artificial Intelligence

You can be creative. Experiment and modify to fit your personal use because AN-83 and Eliza both possess source code in basic, the most popular easy to use language for the micro. Their extensive, easy to understand commands walk you through the source code step by step. It's surprisingly simple. Even the beginner can understand the "How and Why of A.I."

A Fantastic Savings

The real profit to Ennon Corporation will be your participation in the future of Artificial Intelligence. Therefore, I am pleased to say nobody will miss this chance because they could not afford it. AN-83 is priced to cover just a fraction of its research and developmental costs.

The "Thinking" Program AN-83 is just $21.57. What's more, the astounding Eliza is yours absolutely free.

I guess it's obvious that I want you to participate in the future of Artificial Intelligence. Forgive my excitement and enthusiasm but I just know you are going to be very happy and impressed that such things could be done with your computer. You just won't believe it. Please take this opportunity now. Simply fill out your coupon below and mail today. Don't miss out. It's such a wonderful future of discovery and excitement that awaits you.

With very best of wishes,

Vincent Kurek

Vincent Kurek

....

☐ Please send me the "Thinking" Program AN-83 for only $21.57. In addition, I will receive absolutely FREE Eliza—the most impressive conversational Artificial Intelligence program to date.

Available in disk only.
For use with the Commodore 64.

Name ______________________

Address ______________________

City ______________________

State ________ Zip Code ______

THE ENNON CORPORATION
1817 W. Call St., Suite B-8
Tallahassee, FL 32304

Canadian orders must be in U.S. dollars.
We export to other countries.
Commodore is a registered trademark.

example, and you'll see that the machine language instruction A9 is used twice: once in Line 1, and again in Line 3. Now look at the source code listing, and you'll see that the assembly language mnemonic LDA is also used twice, on the same lines and in the same positions as the machine language instruction A9. You might guess then, that the object code instruction A9 equates to the source code instruction LDA. And, as it turns out, that's true.

Take another look at the object code listing, and you'll see that the machine code instruction 20 is also used twice And in both cases, it can be equated to the source code instruction JSR. And so on.

Now you've had a firsthand look at how assembly language compares with machine language. Even though you may not yet understand either language, you've seen that there is some kind of close correlation between the two. And that's all I wanted to get across in this example. So now we can put the topic of machine language on hold for awhile, and focus on the assembly language version of the *HI.TEST* program:

HI.TEST.SRC
(HI.TEST Program, Source-Code Version)

Col. 1 Line #	Col. 2 Source Code	
1	LDA	#72
2	JSR	$FFD2
3	LDA	#73
4	JSR	$FFD2
5	RTS	

WHAT THE PROGRAM DOES

We'll start our examination of this listing by looking at Column 2—the column that contains the hexadecimal number FFD2 and the decimal numbers 72 and 73.

In Commodore assembly language, the number 72 is a screen display code that equates to the letter H. The number 73 is a display code number for the next letter in the alphabet, the letter I. And the hexadecimal number FFD2 (65490 in decimal notation) is the starting address of a handy machine language subroutine that's built into the Commodore 64: a subroutine that will print a character on the screen.

In the *HI.TEST.SRC* program, the numbers 72 and 73 are preceded by the symbol "#," and the next number FFD2 is preceded by the symbol "$." Here's what those symbols mean:

When the symbol "#" precedes a number in Commodore 64 assembly language, it means that the number is to be interpreted as a literal number, not as a memory address. In the *HI.TEST* program, if the numbers 72 and 73 were not preceded by the symbol "#," they would be interpreted as addresses in your computer's memory. But, since they do have a "#" prefix, they are interpreted as actual numbers.

The other special symbol in the *HI.TEST* program—the dollar sign in front of the number FFD2—is an assembly language prefix for hexadecimal numbers. If you're familiar with hexadecimal notation, you can probably tell by looking at the number FFD2 that it's a hexadecimal number. But sometimes decimal numbers and hex numbers look exactly alike. So, in the *HI.TEST* program, the "$" symbol is used to show that the number $FFD2 is to be treated as a hexadecimal number.

Please note, however, that the other symbol in Column 2—the symbol "#"—is *not* used in front of the number $FFD2. In this program, $FFD2 is supposed to be interpreted as a memory address, not as a literal number. So it is not preceded by the symbol "#." In the Commodore 64, as I've mentioned, $FFD2 is the memory address of a built-in subroutine that prints a character on the screen. And that's the subroutine that is called in Lines 2 and 4 of the *HI.TEST.SRC* program.

ASSEMBLY LANGUAGE MNEMONICS

Column 1 of the *HI.TEST.SRC* program contains three assembly language instructions: the mnemonics LDA, JSR, and RTS. We'll now examine each of these mnemonics with the help of a line-by-line analysis of the *HI.TEST.SRC* program.

❶ LDA #72

As I mentioned earlier in this column, when you write a program in assembly language, what you're actually doing is programming the 6510 chip, your computer's main microprocessor. So, before you can start programming in assembly language, you'll have to know a few important facts about your computer's CPU.

Inside your Commodore's 6510 chip are several *internal registers.* You can store data in these registers in the same way that you store data in the memory registers in your computer's ROM and RAM. But the internal registers in the 6510 chip have some special features that ordinary memory registers do not have. The functions and features of the 6510's internal registers will be covered in detail in later columns. But before we go any further, I'll have to say a few words about one specific 6510 register: the *accumulator.*

The accumulator is the busiest register in the 6510 chip. Before any mathematical or logical operation can be performed on a number in 6502/6510 assembly language, the number first has to be loaded into the accumulator. And the assembly language instruction that is usually used to load a number into the accumulator is *LDA.*

In Line 1 of the *HI.TEST.SRC* program, the statement "LDA #72" means "Load the accumulator with the literal number 72." In Commodore 64 assembly language, as I've mentioned, the number 72 is a screen code for the letter "H." So, what Line 1 of the *HI.TEST.SRC* program really means is "Load the accumulator with the screen code for the letter 'H.' "

❷ JSR $FFD2

In 6502/6510 assembly language, the mnemonic *JSR*

GPC

GRAPHIC PRINTER CABLE

The Graphics Printer Cable (GPC) by Q-R&D is the truly universal graphic parallel printer interface for the Commodore 64™, SX-64™, and VIC-20™ computers.



GPC Features:

- ☐ For Virtually All Centronics-Parallel Dot Matrix Printers
- ☐ Direct and Complete Commodore™ Printer Emulation, Graphics, and Software Compatibility for most popular dot matrix printers including Epson ™, Star Micronics™, Riteman™, Okidata™, Prowriter,™ Banana™, Seikosha™, and most similar printers.
- ☐ External Printer Selection Switches – for easy access when choosing your printer type.
- ☐ Software Configurable – allowing Commodore™ printer emulation, graphics, and software compatibility for other new or non-standard type matrix printers.
- ☐ An Intelligent 2K Buffer Is Standard – which greatly increases printing speed for both graphics and text and allows you to use your computer sooner while printing.
- ☐ Command Channel With More Than 20 Commands – enabling you to control features like printer mode, graphic and control character listing method, text spacing, text width, pagination, character set, and interface reset without interfering with listings or software operation.
- ☐ Cardco™ Print Mode Compatible – The GPC is compatible with software written for use with printers using Cardco™ interfaces as well as Commodore™ printers, insuring maximum software compatibility.
- ☐ Complete Users Manual – including software examples and Hi-Res machine language screen dump.

QR&D

ONE WEST LAKE STREET SUITE 320, MINNEAPOLIS, MINNESOTA 55407 **612 922-0516**

Reader Service No. 144

means "Jump to subroutine." This instruction is used in much the same way as the GOSUB instruction is used in BASIC. When the mnemonic JSR is used in an assembly language program, it causes the program to jump to a subroutine that is expected to start at whatever memory address follows the JSR instruction.

In assembly language, the mnemonic JSR is usually used along with another mnemonic, RTS. RTS means "Return from subroutine."

The RTS instruction also corresponds to a BASIC instruction: RETURN. When a JSR instruction is encountered in an assembly language program, here's what happens: first, the address of the very next instruction in the program is placed in an easily accessible location in a special block of memory called a *hardware stack.* Then the program jumps to whatever address follows the JSR instruction. This address is usually the starting address of a subroutine.

When a subroutine is called with a JSR instruction, it is usually expected to end with an RTS instruction. When that RTS instruction is reached, any address that has been placed on the stack by a JSR instruction is retrieved. The program then returns to that address, and processing of the main body of the program resumes.

So, in Line 2 of the *HI.TEST.SRC* program, then, the statement "JSR $FFD2" means "Jump to the subroutine that begins at Memory Address $FFD2." And, as I mentioned earlier, Memory Register $FFD2 is the starting address of a handy machine language subroutine that's built into the Commodore 64. This subroutine takes whatever screen code is stored in the accumulator and automatically prints the character that corresponds to that code on the screen. Then it returns control to whatever program is in progress: in this case, the *HI.TEST* program.

A number of handy I/O routines that work much like this one are built into the Commodore 64. And we'll be using quite a few of them in this series of columns.

❸ LDA #73

In Commodore 64 assembly language (and Commodore 64 BASIC, too), the number 73 is a screen code for the letter "I." So, in the *HI.TEST.SRC* program, the statement "LDA #73" means "Load the accumulator with the screen code for the letter I."

❹ JSR $FFD2

This statement is identical to the statement in Line 2. It means "Jump to the subroutine that starts at Memory Address $FFD2." This time, however, since the accumulator has been loaded with the value 73, the subroutine that starts at $FFD2 will cause an "I" to be printed on the screen.

❺ RTS

RTS, as I have already explained, is an assembly language mnemonic that means "Return from subroutine." When an RTS instruction is used to terminate a subroutine, it usually causes a program to jump back to where it left off before the subroutine was called. In this case, however, RTS is used in a slightly different way: to terminate a whole program, rather than just a subroutine. When RTS is used in this fashion–to terminate a complete program–it usually returns control of a computer to whatever program or system was in control before the program began. So, if you were to call the *HI.TEST* program from BASIC, the RTS instruction in Line 5 would transfer control to the BASIC interpreter.

CHOOSING AN ASSEMBLER

Before an assembly language program can be executed, it is necessary to convert (or *assemble*) it into machine language. It is possible to assemble a program by hand–in fact, that's the way that all programs were assembled before automatic assemblers came along. But the easiest way to assemble a program is with a special kind of software package called an assembler.

You won't need an assembler to convert the *HI.TEST* program into machine language, since I've already done that for you. But, to type and run most of the programs that will be presented later on in this series of columns, you'll need access to a machine language assembler. So I strongly suggest that you buy an assembler before *Ahoy!* publishes next month's column.

There are many assemblers on the market. But to write the programs that will appear in this series of columns, I will use only one: the *Merlin 64* assembler, manufactured by Roger Wagner Publishing, Inc. (formerly Southwestern Data Systems) of Santee, CA. I wouldn't dream of telling you what kind of assembler you should buy, but if you want to type, assemble, and run the programs that will be presented in these columns, it wouldn't hurt to have access to the assembler that they were created on. So, if you don't own one already, you might want to consider buying a *Merlin 64.* (For more information on this program, see the review on page 26.)

OTHER SUPPLIES

When you visit your Commodore dealer to buy your assembler, you might also take a look at a few other supplies that may help you in your study of assembly language. I'll assume that you already have the most important piece of equipment, a Commodore 64 computer. And, since most of the utility packages used in assembly language programming are disk-based, you'll also need a C-64-compatible disk drive.

A Commodore 1520 or 1525 printer, or any other type of line printer that's compatible with the C-64, will also come in handy. It doesn't have to be a letter quality printer, but it should be capable of printing out readable listings of assembly language programs. Another useful (if not essential) item for Commodore assembly language programmers is the official *Commodore 64 Programmer's Reference Guide,* published by Commodore and distributed by Howard W. Sams & Co., Inc.

Continued on page 90

BASIC Trace

For the C-64

By Daniel M. Green

BASIC Trace is an all-machine language utility designed to help you discover which lines your BASIC program is executing at any given time. The program creates a window in the upper right hand corner of the screen which displays the last five line numbers of your BASIC program that have been executed. As soon as your BASIC program executes a new line, that line number is added to the bottom of the trace window. By watching the line numbers march across the window, you can easily determine the flow of a BASIC program.

HOW TO USE THE PROGRAM

BASIC Trace contains a BASIC program which loads a machine language routine into a free block of memory. You must use our *Flankspeed* program to enter the ML portion of *Trace;* see page 94. The ML is put into locations 49152 through 49659. Since BASIC can't "see" these locations, the ML is relatively safe there.

Type in the program, taking care to get the data entered correctly. SAVE the program and RUN it. It will print a brief set of directions on the screen, and then pause as the machine language routine is POKEd into memory. If the program prints "ERROR IN DATA STATEMENTS", you typed some of the data in wrong. Check the program listing carefully, correct the mistake, reSAVE the program, and RUN it again. If all is well, it will respond with the following directions:

```
BASIC TRACE
TO BEGIN TRACING YOUR BASIC PROGRAM,
TYPE:   TRACE <RETURN>
THE COMPUTER WILL PRINT 'ON', AND THE
TRACE WINDOW WILL APPEAR.  NOW RUN YOUR
BASIC PROGRAM.  TO SPEED UP THE TRACING
HOLD DOWN THE SHIFT KEY.
WHEN YOU NO LONGER WANT TRACE, TYPE:
TRACE <RETURN>. THE COMPUTER WILL PRINT
'OFF', MEANING TRACE IS NOW OFF.
TO RESTART THE TRACE, TYPE:
TRACE <RETURN> AGAIN.
```

Once you see these messages, the machine language *Trace* routine has been successfully stored in memory.

The purpose of *BASIC Trace* is to examine the operation of a BASIC program. Therefore, you should now NEW this loaded program, and LOAD in the program you wish to monitor. After you LOAD it, but *before* you RUN it, type:

```
TRACE <RETURN>
```

The computer will print "ON", signifying that TRACE is now running. Next, it will create a *window* in the upper right hand corner of the screen. To change the color of the window, just POKE 49194, color (where color is a number from 0 to 15). Now you are ready to begin. Type RUN <return> to start your BASIC program going. You will see the line numbers of the program scrolling across the trace window. At the top of the window is the word "PAUSE". This indicates that *Trace* is pausing for about one second before executing the next line of your BASIC program. The reason it pauses is so you can clearly see the line numbers in the window. However, if you wish to "fast forward" through your program, hold down the SHIFT key. Now the line numbers will fly across the window. The word "<RUN>" appears, telling you that your program is running at its maximum speed. If you wish to slow it down again, so you can see the line numbers, just let go of the SHIFT key. Note that if you press the SHIFT-LOCK button the *Trace* will "<RUN>" without your having to keep a finger on the SHIFT key.

After you have traced through your program, you might want to disable *Trace.* To do this all you must do is type:

```
TRACE <RETURN>
```

The computer will print "OFF", signifying that tracing is no longer in effect. If you press SHIFT-CLR/HOME, the trace window will disappear. If you decide you want *Trace* again, just type *Trace* <return> and the C-64 will print "ON" and reactivate the window. Now, if you really want to kill *Trace*, you must type: SYS 64738 <return>. This is the system reset command, and will wipe out *Trace,* along with the BASIC program you're currently working on. If, after doing this, you realize that you want *Trace* back, you can type SYS 49152 <return>. This will relink *Trace* to the BASIC interpreter, provided of course that you haven't overwritten locations 49152-49659.

THE *TRACE* WINDOW

The *Trace* window is refreshed only when either (a) you press the RETURN key, or (b) when a BASIC program is running. Thus you might type TRACE <return>; the computer would print "ON", and then you might press CURSOR DOWN dozens of times. This will scroll the screen upwards, and the trace window will disappear from view. To see the window again, just hit the RETURN key, and the window will reform in its usual place.

Continued on page 92

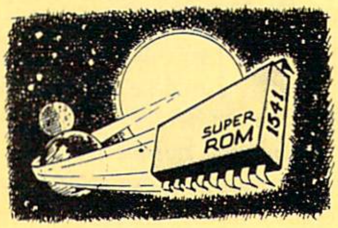
A Survival Guide to the 1541 Disk Drive

This is a 40 + page book and comes complete with all commands and examples on how to use them. **$6.99**

The Complete Book on Telecommunications **$6.99**

A 40 + page book that will help you get started with your Modem. How to use it and how to access a BBS. Compuserve, Dow Jones as well as up and down load.

1541 Super ROM **$39.95**



- *Fast Save Load Verify*
- *Fast Scratch and Validate*
- *10 Second Format with Verify*
- *No more Drive Head rattling during Format or Error Reading*

also Save with Replace is Improved

- *Two times faster, Eight times faster when used with Turbo 64 which is included*
- *1541 Super ROM is 100% Compatible*
- *Easily installed in Minutes*

APALLO COPIES IT ALL



APALLO Does it All. This program is the latest generation of copy programs. It will do everything the $39.00 and $49.00 programs will do and **more.** It Copies **ALL** drive errors, bad tracks and sectors, non-standard format, bad syncs, and half tracks. We feel this is the best program of its kind available...

$29.95

MSD Sure Copy

At last a complete utility package for the MSD Dual Drive. ***This is the first MSD utility program that does it all.*** *The main menu options include:*

- *Copy Protected Disk*
- *Copy Files*
- *Format a Disk*
- *Change Disk Name*
- *Quit*
- *Copy Unprotected Disk*
- *Scratch a File*
- *Rename a File*
- *View Directory*

$39.95

Sure Copy *will put all errors automatically on disk: 20, 21, 23, 27 and 29's.*

D-CODER

- Translates any machine language program into easy-to-read English descriptions with complete explanations of each command!
- Makes complete notations of all important memory locations accessed by the program! (SID, VIC, MOS, KERNAL, etc.)
- Gives you three ways of accessing programs:
  1) **Will read and list programs from DISK**
  2) **Will read and list programs from MEMORY!**
  3) **Direct user input** (from magazines, etc.)
- Can be used to locate and examine any machine language program's protection routines!
- Can be used to easily break apart machine language programs for study and examination!
- Printer option for complete hard copy listings!

$19.95

You no longer need to be an EGGHEAD to read Machine Language.



N-CODER **$19.95**

the machine language manipulator...



The perfect companion program to D-CODER!

Allows you to easily make changes in machine language programs... right on the disk!

- *Rewrite ability allows code to be altered and then rewritten directly to the disk!*
- *Features sector-by-sector scrolling assembly language display of machine language programs!*
- *Notation of ASCII text equibalents for easy spotting of embedded text strings!*
- *Handy reference display of all assembly language commands and their ML numerical equivelents!*
- *Byte splitter for easy splitting of decimal addresses into low byte-high byte format!*

C-64 Label Maker

Choose one to one million labels...

$14.95

- On-line Instructions
- Allows Fall Editing
- Works with all Standard Printers
- This Program will even put your Disk Directory on Labels for you

Enclose Cashiers Check, Money Order, or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders. Canada orders must be in U.S. Dollars. VISA — MASTER CARD — C.O.D.

Programs for C-64 $2.00 S & H on all orders.

Software Submissions Invited.

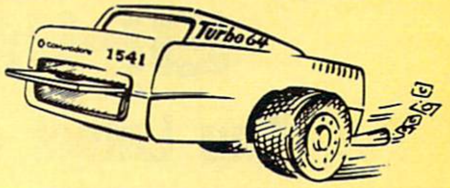
MegaSoft Limited

P.O. Box 1080, Battle Ground, Washington 98604

Phone (206) 687-5116 • BBS 687-5205 After Hours Computer to Computer

TURBO 64

$19.95



Turbo 64 will turn your 1541 into a super fast and efficient disk drive. Loads programs five to eight times faster, works with 99% of your basic and machine language programs. The master disk allows you to put unlimited Turbo 64 boot copies on all your disks. This disk also includes:

Auto Run Boot Maker

Auto Run Boot Maker will load and auto run your basic or machine language programs.



$19.95

DMS-Errors 20,21,22,23,27 & 29 Format Single Tracks Read Disk Errors

½ Track Reader-read and select ½ track.

½ Track Formatter-Format a disk with ½ tracks. This is where the next protection schemes are coming from.

Drive Mon-Disk Drive assembler/disassembler. For your 1541.

The Doc-Disk Doctor that reads code under errors.

Sync Maker-Place a sync mark on **any** track out to 41. Also used for protection.

Sync Reader-Check for Sync bits on **any** track out to 41.

Change Drive No.-Changes drive number (7-30).

Disk Logger-Finds starting track, sector; start and end addresses.

Disk Match-Compare any two diskettes. Byte for Byte.

New Wedge-Easier to use DOS wedge.

ID Check-Check ID's on **any** track.

Unscratch-Restore a scratched file.

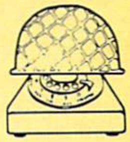
View RAM-Visual display of the free and used sectors on a diskette.

Read/Write Test-1541 performance test.

Repair a track-Repair a track with checksum errors. Reads code under errors and restores track.

Fast Format-Format a disk in just 10 seconds (with verify!).

WAR GAMES AUTODIALER



1-**Auto Dial** will automatically dial a set of numbers you choose.
2-**Review Numbers** will review numbers that were answered by a computer.
3-**Save Numbers** will save numbers where a computer answered.
4-**Hardcopy of Numbers** will print out list of numbers where a computer answered.
5-**LOAD Numbers** will load in numbers to continue where it left off.
6-**Continue** will pick up dialing where it was interrupted.

$29.95

KARTRIDGE KRACKER



NOW you can own this unique and powerful tool which will allow you to dump the contents of 8K and 16K cartridges onto disk! But what's really great is that you can also RUN the cartridges programs without plugging in the cartridge! the KRACKER gets YOU INSIDE the cartridge! Put all your favorites on disk and get rid of the clutter. This package provides your with the software and hardware needed to get started. Program on disk included. (Some cartridges require use of external RAM not included)

$44.95

SOFTWARE PROTECTION HANDBOOK

3rd Edition

THIRD EDITION! NOW AVAILABLE!

If you're **tired** of being harassed by **protected software** and too many **copy** programs, then this is the book for you! This 224 page manual covers the gambit from legalities to protection methods to step-by-step back up procedures. Now you can learn both how to **protect and unprotect** software! The techniques covered include copying cartridges to tape or disk, tape protection, and disk protection. Disk protection covers error no.'s 20, 21, 22, 23, 27 and 29 **plus** single track formatting, header modification, header swapping, half track reading and writing, reading and writing modified bit densities, formatting illegal tracks/sectors, sync writing and **more!** The Third edition **explains**, tells how to **detect** and how to **write** them with included software. Eleven useful utilities and many protection listings! Our disk analysis programs reveal the protection methods used on your originals. A diskette with all software is available for a minimum extra charge. This may not be the only book your should have for the C-64, but it is certainly **the one** book you should not be **without!**

C64 Book only	$19.95 US
Book & Disk of all programs	$29.95 US
Vic 20 book . . . Cart. & Tapes only	$9.95 US

THIS MANUAL DOES NOT CONDONE PIRACY

*SHIPPING: $2.00

The Original Book of Facts

UNLOCK THE MYSTERIES OF THE COMMODORE

—Simple Instructions
—Easy step-by-step examples
—Tricks you can do your first day
—Introduction to color, sound and much more
—Includes a routine to protest your own programs
—NO MATTER what your experience, the Original Book of Facts is for everyone!

$6.99

The Advanced Book of Facts

EXPLORING YOUR COMMODORE

—How to use a machine language monitor
—Tips on how to unprotect your software
—Advanced protection for your own programs
—Full size reference charts
—How to design your own menus and more...

$6.99

BULLETIN BOARD

Set up and operate your own bulletin board with one or two disk drives. This one has all the features and you can customize it easily yourself.

1-RUN MEGASOFT BBS
2-CREATE MEGA FILES
3-ADD TO SYSOP'S CORNER
4-NEW SYSOP'S CORNER
5-READ MESSAGES
6-SCRATCH MESSAGE
7-CYCLE MESSAGES
8-READ SYSOP MESSAGES
9-WRITE OPENING MESSAGE
10-READ LOG
11-CYCLE LOG
12-READ DOWNLOAD FILE
13-SCRATCH DOWNLOAD FILE
14-ADD TO OTHER SYSTEMS
15-CREATE OTHER SYSTEMS

$59.95

Enclose Cashiers Check, Money Order, or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders. Canada orders must be in U.S. Dollars. VISA — MASTER CARD — C.O.D.

Programs for C-64 $2.00 S & H on all orders.

Software Submissions Invited.

VISA

MegaSoft Limited

P.O. Box 1080, Battle Ground, Washington 98604

Phone (206) 687-5116 • BBS 687-5205 After Hours Computer to Computer

Reader Service No. 148

SHIP TO SHORE

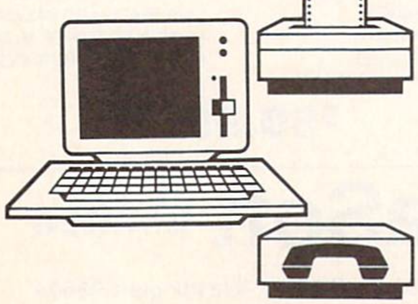
The Latest on the Telecommunications Front

By Cheryl Peterson

Big corporations are not the only ones having problems with crashers. Smaller Bulletin Board Systems (BBS's) are where the "crackers" cut their teeth. After learning to bust the local boards, it's only a small step up to the local bank. Of course, not everyone is capable of making $100,000 withdrawals from someone else's account using a phony password.

At a meeting for system operators (SYSOPs) and other computer users in the South Florida area, the FBI stated that only crimes involving more than $7000 were worth investigating. They also said they were involved with all kinds of crimes using computers. These range from illegal access to public message systems to obtaining phone service passwords and numbers, such as Teltec or Bell credit card numbers.

In one instance, a bill of $40,000 was run up on a Sprint number. The investigation revealed that the necessary information to use the card had been posted on a local public access bulletin board. The user logs of several BBS's were confiscated by the FBI. Most SYSOPs cooperated without a search warrant being necessary, just to clear themselves.



In an effort to police themselves, several area SYSOPs got together a couple of years ago and formed the South Florida SYSOPs Association. A SYSOP's board was set up for messages between members. Any problems that a SYSOP had would go up. If others were experiencing similar problems, action could be taken. Seldom has this been necessary.

All members have messages posted on their boards warning against tampering or leaving messages that are in poor taste or of questionable legality. They agreed that anyone on any board who violated the Florida Computer Crime Act would be banned on all SFSA member boards. Each individual SYSOP would handle problems on his board. Only in cases where someone caused trouble on several or did something illegal would they be banned. To date only three people have been subjected to this treatment. Most association business is conducted on their SYSOP bulletin board.

Meetings are held infrequently. Usually these are parties for prospective members. There are some requirements for membership. In order to be considered, a SYSOP has to have been operating his board more or less constantly for a year. This is to prevent fly-by-night boards from becoming involved. Many boards go up one night and down the next. All members agree to abide by the association's rules.

It seems to be working. With rare exceptions, the boards within the association experience few problems.

Another group formed in Palm Beach County hasn't been so well organized. Its nine members tend to take action before investigating all the facts. Problems aren't discussed as effectively. Banning people is done indiscriminately. At one time, a founding member of the group, Karl Meyer, found himself banned from several other boards. He had asked a SYSOP who was antagonizing some of his users to desist. When the behavior didn't improve, Meyer kicked the offensive SYSOP off his board. Within a few days, Meyer found he'd been devalidated on five boards. No warning, no discussion, nothing. An offense against any board, whether out of malice or inexperience, sometimes brings swift punishment. The immaturity of the group seems to be a real problem.

One important contribution that comes out of these associations is the improvement of the boards. Since many operators are also programmers, sharing their secrets of keeping unauthorized individuals out can really help. Traps set in the BBS program can catch potential "crackers" before they can do any damage.

For instance, many crackers will try entering extremely long names in order to crash the computer into BASIC. Once there, an experienced hacker can cause havoc. One trap tests for lengthy names. If the name is strange, it prints a message that some people have weird names and hangs up.

Another trick is to enter in too many files. This will also cause the system to crash into either BASIC, or worse, the disk operating system. Again, this opens the system up to tampering. One way to prevent this is to limit the number of files any user can create. A few lines of code

Continued on page 84

The Joy of Sticks

Continued from page 22

house—my Atari, my PCjr, my VIC, and my Commodore—will become scrap.

I'm not holding my breath.

But it *might* happen, right? So here it is, the ideal home computer from my point of view.

THE PERFECT HOME MACHINE

ROM. The entire operating system will be in ROM, present whenever the computer is powered up—like the 64, and unlike all the other machines. There will also be a BASIC and a screen editor in ROM—unlike the Adam and PCjr. And no ROM will use up RAM space or involve tedious switching in and out. (This means that, alas, the old 6502 and Z80 processors cannot possibly be used in the perfect machine. Neither, for that matter, can the PCjr's pseudo-16-bit processor. I want *no* bank-switching.)

Read/write Memory. There will be Enough. I'm not saying how much that is, because it depends on how the graphics and operating system are handled, but when you buy the basic machine, you will have Enough Memory. And it will all be in a neat, orderly arrangement so you don't have to do some of the contortions we did last month to fit a complex graphics game into memory.

Graphics. It will look good on a home television set. Everything will be programmable, both through direct manipulation of memory locations and through the BASIC language. Characters can be redefined in monochrome and four-color modes. Sprites in one or three or more colors can be moved independently of the background. Screens, character sets, and sprites can be "flipped" for animation. There will be various levels of resolution of bit-mapped graphics. There will be a high-resolution 16-color bit-mapped mode—which means there'll have to be a lot of memory! Hue and brightness will be separately controlled with many possible gradations, so fine shading will be possible.

Sound. There will be enough voices to do complex chords: at least five simultaneous tones. Each voice will be separately programmable, with sound at least as good as the SID chip in the Commodore 64. And sound production won't use up much central processor time.

BASIC. The built-in language will be BASIC, and the BASIC will fully implement all the sound and graphics and peripherals. Text and bit-mapped graphics can be anywhere on the screen at the same time, and BASIC can draw ellipses and polygons and fill them with solid or patterned colors at least as well as the PCjr does. Strings can be of any length, so that whole screen displays can be stored in string arrays and PRINTed into memory as easily as the Atari does. And anyone who produces a BASIC for a home computer that doesn't include humanely designed graphics and sound mini-languages like the DRAW and PLAY commands in PCjr BASIC should be forced to spend the rest of their lives POKEing sound commands into a Commodore 64.

Storage. Fast. Reliable. Easy to use. Hard to mess up. That undoubtedly means disks, with a well-designed disk operating system; but it could also mean bubble memory or battery-powered CMOS RAM—I would love that. It does not mean tape.

Cartridges. There must be a slot for ROM cartridges that can hold complex programs—up to 128K. This still remains the best way to market commercial software, because it is automatically copy-protected, and if your commercial software is entirely on cartridge, you only need one storage device.

You get the picture, don't you? A computer that is designed so that the producers of commercial software can sell you absolutely magnificent programs to plug in and run, and yet is so programmable that children don't have to do more work than a Ph.D. in order to learn to create their own programs.

The latest fashion in computer theory says that home computers don't have to be programmable anymore. Computer users just want software they can buy and run—they don't want to develop their own programs.

Well, the people who say that are the same people who sneer at BASIC because it's an "unstructured" language—you can do unpretty things with BASIC. You can get a D in neatness. What they don't realize is that the computer is supposed to make us free. Like the VCR, which lets us determine our own viewing schedule and even make our own TV programs, the computer is *not* in my home so that I can have only the programs that some company thinks at least 100,000 people will buy. There are sometimes things that only one person will buy—me. And I, for one, will own a computer that lets me create that program.

It's like traveling from coast to coast. Sure, most of the time I'd rather fly. But sometimes I like to get in the car and drive. And not on freeways, either. Sometimes getting there *is* the fun. Besides, the planes only fly to the places where *everybody* wants to go. Sometimes I want to go where no one has gone before.

So I won't buy a computer that doesn't let me program. Computers that don't let me do more than what I can buy from software vendors are, in my opinion, no better than souped-on versions of the Atari 2600. Not a *real* computer at all.

A REVIEW OF SCREEN MOVEMENT

In case you're one of the many thousands of readers who have started reading *Ahoy!* in the last six months, here is a brief review of the principles of moving a figure around on the screen.

The figure, in this case, is a single character, held in the variable PF$. To make it seem to move, we must put it on the screen, then erase it at the old location and put it at the new location. We do this over and over again: erase it at the old place, put it at the new place. If we do it fast enough, and in small steps, it looks to the human eye as if the *same* figure has actually moved.

Erasing is easy: We PRINT a blank space at the old location. Then we PRINT the figure, PF$, at the new

location.

What's tricky is keeping track of where the location *is*. Think of the TV or monitor screen as a grid. There are 25 possible vertical positions for the figure, and 40 possible horizontal positions. Because of the way the 64 handles lines and scrolling, we won't use the rightmost column or the bottom row, so that our program will act as if there were only 24 possible vertical positions, numbered from 0 to 23, and only 39 possible horizontal positions, numbered from 0 to 38. (With the VIC, of course, there are fewer rows and columns; the VIC version of the example program shows the numbers.)

The variable H is used to hold the horizontal position of the figure; the variable V holds the vertical position. The variable HX holds the *old* horizontal position, and VX holds the old vertical position. So we can erase the old figure and draw the new with a single statement:

```
PRINT V$(VX)TAB(HX)" "V$(V)TAB(H)PF$;
```

To use this statement, we first have to set up a string array, V$*(n)*, with one string for each of the 24 possible vertical positions. V$(0) consists of the *home* character, so that PRINTing V$(0) moves the cursor to the top left corner of the screen. So the top row, row 0, is reached by PRINTing V$(0). Each subsequent V$*(n)* string consists of a *home* character followed by as many *cursor-down* characters as the number of the string. That is, V$(23) consists of *home* and 23 *cursor-downs*. This puts the cursor on row number 23, which is the lowest position we can use on the 64 screen.

Then the TAB function, combined with the horizontal position variable H, puts the figure in the correct left-right position. If H has a value of 0, the figure will be in the leftmost column; if it has a value of 38, the rightmost column.

When the player calls for the figure to move, H and V are changed to the new positions for the figure. Say the player wants to move to the right. The value of V is not changed, since it will stay on the same row. But the value of H will be *increased* by 1, which will move the figure one space to the right. The values of HX and VX are not changed. So executing the above statement causes the old figure to be erased and the figure appears at the new location, one space to the right. Only then are HX and VX set to the values of.H and V, so they will now represent the current position of the figure when it's time to move again.

If the player calls for a movement that would cause H or V to be less than 0 or greater than the number of the rightmost column or bottom row, the program has two choices. Either the movement instruction is ignored, which causes the figure to reach the edge and stop, or the movement instruction causes the player to "wrap around" to the opposite edge of the screen. This means that if H has a value of 0 and the player calls for it to move left, the figure will now be PRINTed in the *rightmost* column, where it will continue moving left as long as the player wants it to.

The variables BE and RE hold the value of the bottom row and rightmost column allowable. The variables BF and RF are one more than BE and RE: these are the numbers added to or subtracted from H and V in order to wrap around to the opposite edge of the screen.

In both the VIC and 64 example programs, the lines from 100 to 190 carry out these tasks:

Read the joystick.

Check to see which movement direction is called for, if any.

Check to see if a movement causes the figure to move off the screen, in which case the figure is wrapped around to the other side.

PRINT a blank at the old position and the figure at the new position.

Set HX and VX to the new values of H and V.

Go back and do it all again.

That's screen movement in a nutshell. Every computer movement scheme does exactly the same tasks, with only slight variations. For instance, if you are moving across a background, as we were in last month's program, the movement loop must read and remember what background character was in a certain position before the figure moved to that spot, and then put that character back when the figure moves on. Another variation comes with sprites—one of their chief virtues is that you don't have to erase them at the old position, and the background under them remains undisturbed. But the basic pattern of keeping track of movement remains the same. ☐

SEE PROGRAM LISTINGS ON PAGE 96

COMMODARES

PROGRAMMING CHALLENGES

By Dale Rupert

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting, and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Also show sample runs if possible, where appropriate. Programs on diskettes are welcome, but they must be accompanied by listings. Also tell what makes your solutions unique or interesting, if they are. You must enclose a stamped, self-addressed envelope if you want any of your materials returned.

Your original programming problems, suggestions, and ideas are equally welcome! The best ones will become *Commodares.*

Problem #16-1: Numerical Columns

The user enters C, the number of columns from 1 to 5, and N, the ending number from 1 to 1000. The computer displays C columns of numbers on the screen. All columns are the same length with the possible exception of the last column which may be shorter. The columns are numbered vertically as shown in this example: if C = 4 and N = 11, the output is:

```
1   4   7   10
2   5   8   11
3   6   9
```

What is the simplest way to do this?

Problem #16-2: Quick Decimal

Both James Speers, M.D. (Niles, MI) and John Prager (Bay City, MI) sent with their solutions to *Problem #12-4: Quick Hex* the suggestions to include the opposite problem. Here it is. Write the shortest routine possible to convert any hexadecimal number from 1 to 4 digits to the corresponding decimal value. We will see their solutions next month. Thanks for the suggestion.

Problem #16-3: Time Warp

Here's a little problem meant to encourage those of you who don't know or have forgotten how to do it to open the encyclopedia or a physics book and give it a try. From Einstein's theory of relativity, it is known that, when measured from an object at rest (earth), time on an object in motion (rocket) progresses more slowly.

Write a program that allows the user to enter the speed of the rocket as a percentage of the speed of light for any value up to 100. The computer outputs the speed of the rocket in kilometers per second and miles per hour as well as the number of years that pass on earth for each year on the rocket.

Problem #16-4: Common Pairs

The user enters a sentence of up to 255 characters. The computer displays all sets of two adjacent letters which occur more than once in the sentence. For example, in the sentence "This is his third ride," the two-letter combinations occurring more than once are "th", "hi", and "is". The two-letter combinations "ir", "rd", "ri", "id", and "de" occur only once each.

A letter next to a space is considered to have just one adjacent letter. Only left to right order is considered. Consequently the "ir" in "third" does not match the "ri" in "ride." How simply can you do this?

This month we'll pick up some old business, then look at readers' solutions to the December *Commodares.* As promised, here is David Patterson's (San Jose, CA) solution to *Problem #15-2: Prime Factors.*

```
1 REM PROBLEM #15-2 : PRIME FACTORS
2 REM  SUBMITTED BY DAVID PATTERSON
110 PRINT CHR$(18)"ENTER POSITIVE INTEGE
R TO BE FACTORED"CHR$(146)
120 INPUT X1:IF X1<1 THEN RUN
130 PRINT"THE PRIME FACTORS ARE AS FOLLO
WS:"
150 X2=2
160 GOSUB 300
170 X2=3
180 GOSUB 300
220 FOR N=0 TO SQR(X1)/6
230 X2=6*N+5
```

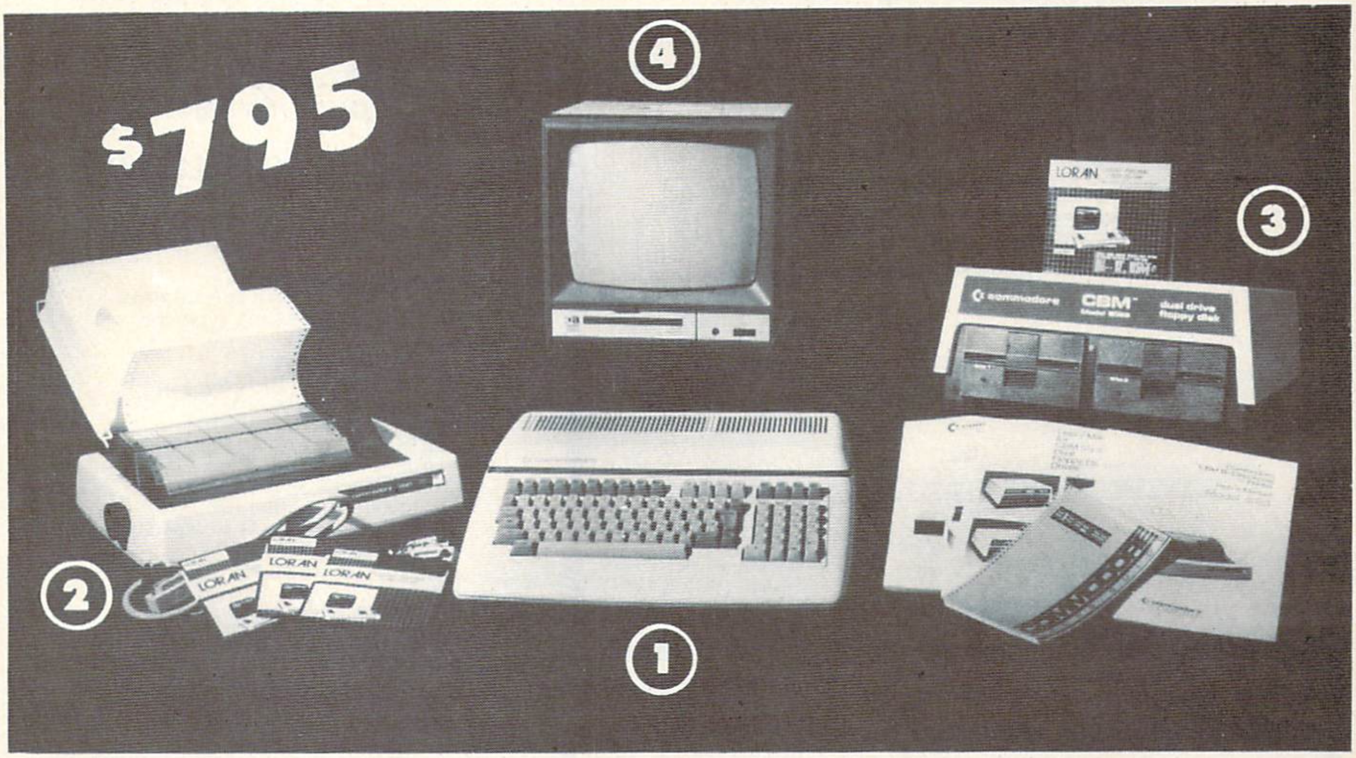
Continued on page 89

Reader Service No. 150

NEW 128K —MEGA BYTE DUAL DISK DRIVE—80 COLUMN

COMPUTER SYSTEM SALE!

HOME • BUSINESS • WORD PROCESSING



LOOK AT ALL YOU GET FOR ONLY **$795.**

	LIST PRICE
① B128 COMMODORE 128K 80 COLUMN COMPUTER	$ 995.00
② 4023 - 100 CPS - 80 COLUMN BIDIRECTIONAL PRINTER	499.00
③ 8050 DUAL DISK DRIVE (over 1 million bytes)	1795.00
④ 12" HI RESOLUTION 80 COLUMN MONITOR	249.00
• BOX OF 10 LORAN LIFETIME GUARANTEED DISKS	49.95
• 1100 SHEETS FANFOLD PAPER	19.95
• ALL CABLES NEEDED FOR INTERFACING	102.05
TOTAL LIST PRICE	**$3717.95**

$795 SALE PRICE

PLUS YOU CAN ORDER THESE BUSINESS PROGRAMS AT SALE PRICES

Programmers Reference Guide
List $29.95
Sale $24.95

	LIST	SALE	*SYSTEM PRICE**
Professional 80 Column Word Processor	$149.95	**$99.00**	***$49.00***
Professional Data Base	$149.95	**$99.00**	***$49.00***
Accounts Receivable	$149.95	**$99.00**	
Accounts Payable	$149.95	**$99.00**	

	LIST	SALE
Payroll	$149.95	**$99.00**
Inventory	$149.95	**$99.00**
General Ledger	$149.95	**$99.00**
Financial Spread Sheet	$149.95	**$99.00**
Order Entry	$149.95	**$99.00**

* When Bought with B128 Computer System.

☆ PRINTER REPLACEMENT OPTIONS

(replace the 4023 with the following printers at these sale prices. Interface Included)

	LIST	SALE
☆ Olympia Executive Letter Quality Printer	$699.00	**$369.00**
☆ Comstar Hi-Speed 160 CPS 15½" Business Printer	$779.00	**$459.00**
Telecommunications Deluxe Modem Package	$199.00	**$139.00**
IEEE to Centronics Parallel Printer Interface	$179.00	**$139.00**

15 DAY FREE TRIAL. We give you 15 days to try out this SUPER SYSTEM PACKAGE!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!
90 DAY IMMEDIATE REPLACEMENT WARRANTY. If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY with no service charge!!

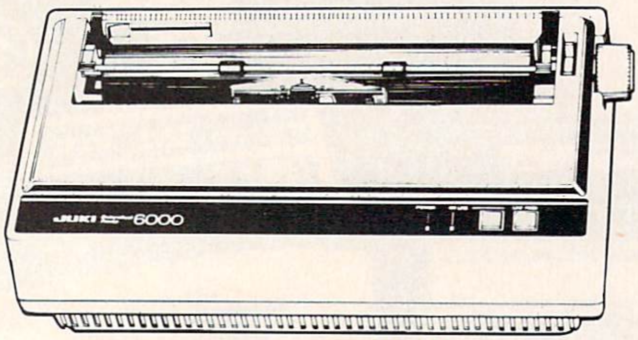
Add $50.00 for shipping and handling!!
WE DO NOT SHIP THIS SYSTEM TO FOREIGN COUNTRIES, CANADA, PUERTO RICA, ALASKA, OR HAWAII.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

DAISY WHEEL PRINTER SALE!

JUKI®

(Japanese Highest Quality Award Winner)



Letter Quality Printer

DELUXE LETTER QUALITY "DAISY WHEEL PRINTERS"

JUKI®

DELUXE COMPUTER PRINTER

List Price $399 **SALE $249**

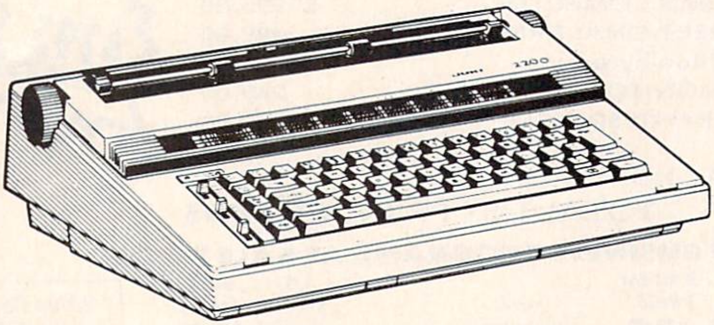
- Superb "Daisy Wheel" Computer Printer
- 100 Characters
- Bi-directional with special print enhancements-many type styl‹ $18.95
- Pitch 10, 12, 15 CPI
- Print Speed up to 12 CPS
- Print line width: 115, 138, 172 characters
- 12" Extra large carriage
- Drop in cassette ribbon (replacement $8.95)
- Centronics parallel RS 232 Serial interface built in (specify)

JUKI®

DELUXE "COMBINATION" PRINTER/TYPEWRITER

List Price $499 **SALE $299**

- Superb Computer Business Printer combined with the deluxe electronic typewriter!
- Two machines in one—just a flick of the switch!
- Superb letter quality corre-spondence—home, office, word processing!
- 12" Extra large carriage
- Drop in cassette ribbon—replacement $8.95
- Precision daisy wheel printing—many type styles! $18.95
- Pitch selector-10, 12, 15 CPS, Automatic relocate key!
- Automatic margin control and setting! Key in buffer!
- Centronics parallel or RS 232 Serial interface built-in (specify)



Letter Quality Printer/Typewriter

15 Day Free Trial - 90 Day Immediate Replacement Warranty

• COMMODORE 64 COMPUTER INTERFACE ONLY $49.00 - ATARI INTERFACE $79.00

ADD $10.00 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD ACCEPTED. We ship C.O.D.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 152

COLOR MONITOR SALE!!!

(Lowest price in USA)



13" Color Computer Monitor

16 COLORS

- Built in speaker and audio
- Front Panel Controls
- For Video Recorders
- For Small Business/ Computers
- Apple-Commodore Atari-Franklin-etc.

- Beautiful Color Contrast
- High Resolution
- Separated Video
- Sharp Clear Text
- Anti Glare Screen
- 40 Columns x 24 lines
- Supports 80 Columns
- List $399

SALE $195

15 Day Free Trial - 90 Day Immediate Replacement Warranty

12" XTRON SUPER HI-RESOLUTION "SWIVEL BASE" MONITOR

List $249 SALE $119 *

80 Columns x 24 lines, Super Hi-Resolution 1000 lines
Green or Amber super-clear Easy to Read text with special anti-glare screen!

12" ZENITH HI-RESOLUTION GREEN OR AMBER TEXT DISPLAY MONITOR

List $199 SALE $99 *

80 Columns x 24 lines, Hi-Resolution-crisp clear easy to read text with anti-glare screen! A MUST for word processing.

12" MONITOR GREEN OR AMBER TEXT

80 Columns x 24 lines, easy to read up front controls

List $159 SALE $79.95 *

***Plus $9.95 connecting cable!**

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

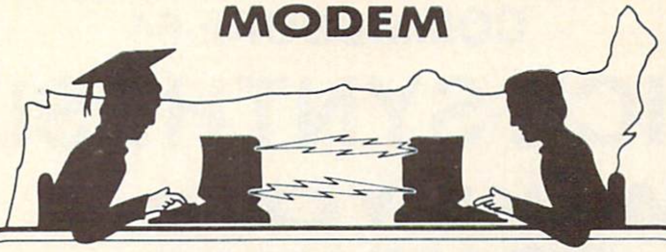
Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Commodore 64
MODEM

Sale



Sale

$59 List $129.95

Telecommunications

$59 List $129.95

FOR CHILDREN·ADULTS·BUSINESS

Complete Auto Dial Telecommunications Package

"The only telecommunications package you will ever need."

(Exclusive Easy To Use Features)

NEW

Total Telecommunications

NEW

- **300 Baud Modem** • **Auto Dial** • **Auto Answer** • **Upload & Download**
- **Membership in 52 Database Services (UPI News, etc.)**

Reach Out and Access Someone

- Educational courses
- Financial Information
- Banking at Home
- Popular Games
- News Updates and Information
- Electronic Shopping
- Research and Reference Materials

The Total Telecommonications Package offers you all this plus ...

- Auto Log-on
- Dialing from Keyboard
- On-line Clock
- Capture and Display High Resolution Characters
- Download Text, Program or Data Files
- Stores on Disk Downloaded Files
- Reads Files from Disk and Uploads Text or Program Files
- Select Any Protocol (access almost any computer or modem)
- Plus Much, Much More

List $129.95 **Special Low-Low Price** **$59.00**

We are so sure this is the only telecommunications package you will need we will give you 15 days Free Trial.

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 155

COMMODORE-64

VOICE SYNTHESIZER

NEW

MAKE YOUR COMPUTER TALK

VOTRAX BASED HARDWARE

SALE ONLY **$59.00**

You can program any words or sentences • Adjust volume and pitch • Receive Modem messages • Make adventure games that talk • Real sound action games • Make customized talkies • (Demo disk or tape included)

You can add TEXT TO SPEECH SOFTWARE that allows you to simply type what you want to hear!! Also allows you to add sound and voice to SCOTT ADAMS and "ZORK" ADVENTURE GAMES. List $29.95. SALE $19.95. (Disk or Tape).

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

COMMODORE 64

80 COLUMN BOARD

SALE

FOR ONLY **$89.00**

(Word Processor Disk Program included FREE!!)



Now you can program 80 columns on the screen at one time! Converts your Commodore 64 to 80 columns when you plug in the PROTECTO 80 Expansion Board. List $199.00. **Sale $89.00.**

Includes 4 Slot Expander and can be used with most existing software!!!

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
Box 550, Barrington, Illinois 60010
312/382-5244 to order

Reader Service No. 157

Commodore 64 | The Most | Commodore 64

Practical Business Software

Special Low Low Prices - While Supplies Last

• Word Processor • Spread Sheet • Data Base • Diagnostic

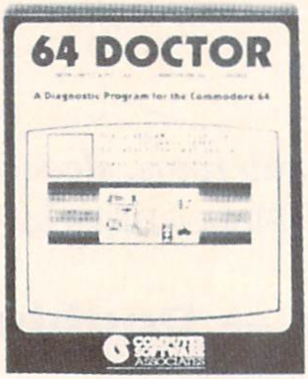
• Track Expenses, inventories, investments • Make Charts and Graphs • Project Profits

• Keep Mailing Lists • Change Records, numbers, methods of filing • Information Retrieval

• Test RAM Memory • RS-232 Port • Keyboard • Video • Audio • Joystick • Printer • Disk Drive • And More!









Practicalc 64: A consistent best seller, Practicalc 64 has become a reference standard among Commodore 64 spreadsheets. With features like alphabetic and numeric sorting and searching, variable comumn widths, graphing and over 30 math functions, this program is an exceptional value. Practicalc 64 also interacts with Practifile, forming the perfect small business bundle.
List $59.95. **Sale $24.95**. (Disk or Tape)

Practifile: Flexibility and large capacity make Practifile the ideal data-base manager for the Commodore 64. (3800 records per file, sorts 5 fields at once.) Files written with the program are compatible with Practicalc 64 and popular word processing programs such as EasyScript, Word Pro, PageMate and PaperClip. Finally, a full-featured data-base at an affordable price!
List $59.95. **Sale $24.95**. (Disk.)

64 Doctor: A special diagnostic program for the Commodore 64, 64 Doctor takes the guesswork out of isolating troublesome hardware problems. The program tests each piece of hardware to pinpoint defects and help prevent costly and time-consuming service calls. An essential program for all Commodore 64 users!
List $34.95. **Sale $19.95**. (Disk.)

PractiCalc II *Better than Lotus 1-2-3* *Coupon $44.95*

PractiCalc II, a fast, versatile spreadsheet with database and word processing features, does away with erasers, broken pencils and a wastebasket filled with scrap paper. PractiCalc II, with its functions and features, has the ability to complete simple and complex tasks. PractiCalc II is flexible enough to be used for checkbooks, alphabetized lists, home budgets and business financial statements. PractiCalc II is the tool of the eighties. List $69.95. **Sale $49.95.** Coupon $44.95. (Disk)

With PractiCalc II you can:

- Use 250 rows by 100 columns
- Use 36 math functions from simple addition to square roots and trig functions.
- Sort alphabetically and numerically.
- Use upper and lower case letters.
- "Hide" columns of numbers for special reports without losing data.
- Create fixed titles of several rows and/or columns.
- Replicate any data in any area.
- Adjust individual column widths.
- Use graphic display option.
- Set global and individual cell formatting.
- Use IF ... THEN statements.
- Edit labels and formulas.
- Format disks from within the program.
- View disk catalog from within the program.
- Write expanded labels up to 88 characters long.
- Insert, delete and move information with ease.
- Start using PractiCalc II within ten minutes of opening the package.
- Consolidate separate spreadsheets for totalling.
- Search for known and variable entries.

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
Box 550, Barrington, Illinois 60010
312/382-5244 to order

Reader Service No. 158

Commodore 64 **CARDCO** Commodore 64

Fulfill All Your Personal & Business Needs "NOW"

Write Now! — Word Processor

Easy To Use Cartridge Based Word Processor for the C-64, With Full 80 Column On-Screen View Mode

The WRITE NOW! word processor provides you with the easiest to use, fastest to learn format available for your Commodore 64 computer. It will give you all the fetures you would expect in a professional word processor and some features not found in any other Commodore 64 word processor.
(Cartridge) List $49.95. **Sale $34.95.** *Coupon $29.95.*

- Program is on cartridge and loads instantly so there is no waste of time loading from unreliable tapes or disks.
- Built in 80 column display allows you to see exactly what you will print including headers, footers, justification, page numbers and page breaks.
- Can send all special codes to any printer, even in the middle of a line without losing proper justification.
- Page numbering in standard numbers or upper or lower case Roman numerals.
- Full string search and search/replace functions.
- Direct unlimited use of previously stored text from tape or disk called from within text during the print out so it uses no memory space and allows creation of unlimited length documents.
- Full scrolling up and down through text.
- No complicated editing modes or commands.
- Multiple line headers and footers.
- Justification and text centering.
- Block copy and delete commands.
- Save text to tape or disk.
- Easy full screen editing.
- Works with any printer.
- Easy to understand instruction booklet.
- 4 help screens included.

Spell Now — Spell Checker

"Spell Now" is a disk based professional spelling checker that interfaces with the "Write Now" word processor for the C-64. Use "Spell Now" to check the spelling in your "Write Now" files. It includes all features of the most expensive spelling checkers on the market. (Disk) List $39.95. **Sale $19.95.**

- 34,000 word dictionary.
- 1,000 word mini-dictionary.
- Unlimited number of supplemental dictionaries (user-definable).
- You see status of spelling check.
- Menu-driven; user-friendly.
- Fully compatible with "Write Now".
- Allows for marking, immediate correction and viewing in context.
- CARDCO, INC.'s Lifetime Guarantee.

Mail Now — Mailing List Software

The "Mail Now" is a disk based full random access data base designed to be used with the "Write Now" word processor. "Mail Now" allows the user to merge an address file into the text of the word processor for form letters. (Disk) List $39.95. **Sale $19.95.**

- Machine language fast sort.
- User-friendly, totally menu-driven operation.
- 10 character comment line.
- User-defined, print format can print one, two or three labels across.
- 2 character category field.
- Supports 600 entries per disk.
- Quick (in memory) sorts by Zip, Category, Last name and State.
- Includes data back-up utility.
- CARDCO, INC.'s Lifetime Guarantee.

File Now — Database

"File Now" is the inexpensive data base you've been waiting for. "File Now" interfaces with the "Write Now" word processor for the C-64. Helps you with data base management of your "Write Now" files and keeps separate data bases for other important information. (Disk) List $39.95. **Sale $29.95.**

- Functions as a deck of 3 x 5 note cards.
- Fields are user-definable.
- Full edit features; add, edit, delete.
- Searching is bidirectional.
- Wild card searching and printing.
- Capacity up to 700 cards per file-disk.
- Fully compatible with "Write Now".
- CARDCO, INC.'s Lifetime Guarantee.

Graph Now / Paint Now — Graphic/Logo Generator

A full graphics package capable of generating line, bar graphs and graphic art designs such as logos and pictures to be used with "Write Now" for the C-64. Save your graph or logo in a "Write Now" compatible file and print it out along with your "Write Now" text file. (Disk) List $39.95. **Sale $19.95.**

- Allows plotting of x,y coordinates from software or keyboard.
- Compatible with "Write Now" files.
- Menu-driven; user-friendly.
- Draw lines, rectangles and circles.
- 3 fill shades; white, gray and black
- Character font editor
- Load or save graphics.
- Optional joystick control.
- CARDCO INC.'s Lifetime Guarantee.

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
Box 550, Barrington, Illinois 60010
312/382-5244 to order

Reader Service No. 159

American Educational Computer

NEW NEW

• Learn at home • Easy To Use • Colorful Graphics • Motivating and Fun • Teacher Designed

All programs were written by teachers and have been classroom validated. These programs teach in a standard classroom format with rewards as the user gets through the various lessons.



(Disk)

US Geography Facts — Has 35 pre-programmed lessons plus an authoring system that lets you create your own lessons. Excellent for teaching and reinforcing factual knowledge of the U.S. After completing two lessons the user earns the right to play a unique and challenging maze game. Fantastic educational value.
List $29.95. **Sale $16.95**.

US Government Facts — 35 lessons plus authoring system included. This program was designed for the following: • Eighth grade American History • Ninth grade Civics • High School electives in government • College refresher courses in government and American History. Fantastic aid to learning about American government.
List $29.95. **Sale $16.95**.



(Disk)

Commodore 64



(Disk)

AEC Spelling — A series of seven educational software programs for grades 2 through 8. It teaches the spelling of 4,000 words most commonly used in writing (98% of most people's writing vocabulary). The word lists used are the result of over 25 years of research into the writing/spelling needs of children and adults. Each grade level consists of one two-sided disk with word lists on one side and study activities on the other. The activities are designed to develop mastery in spelling utilizing a Test/Teach/Test approach. A pre-test is used to find out which words a user cannot spell correctly. These are recorded on a "Words To Study" list. The activities for learning these words are provided. Finally, a post-test is taken by the learner to measure progress. The approach is simple and rewards are built in. List $99.95. **Special Sale — 7 programs $34.95**. *Individual programs $19.95. (Specify grades 2-8).*



(Disk)

Commodore 64



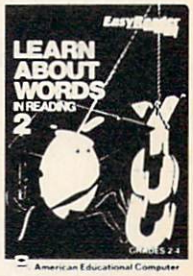
(Disk)

Reading Comprehension Skills — Reading Comprehension Skills, 1, 2, and 3 help the learner develop the ability to read with understanding. The learner does work with cause and effect, similarities and differences, predicting outcomes, finding main ideas, differentiating between fact and opinion, etc.

	List	Sale
Level 1 (grades 1-3)	$29.95	**$16.95**
Level 2 (grades 4-6)	$29.95	**$16.95**
Level 3 (grades 7-8)	$29.95	**$16.95**

Words In Reading (Vocabulary) — Learn About Words In Reading, 1 and 2 teaches the structure of words to ensure success in reading. Such topics as prefixes, suffixes, contractions, synonyms, etc., are among the structural skills included in the program to promote progress and good grades in reading.

	List	Sale
Level 1 (grades 1-3)	$29.95	**$16.95**
Level 2 (grades 2-4)	$29.95	**$16.95**



(Disk)

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

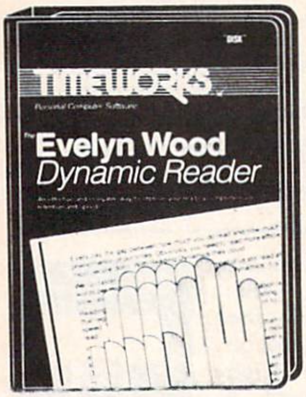
Box 550, Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 161

Com-64 **TIMEWORKS** Com-64

Evelyn Wood Speed Reading



The EVELYN WOOD DYNAMIC READER provides you with the exercises and tools you need to help you increase your reading comprehension and speed. You can use your own personal computer and develop your skills at your own pace. You can learn the essential techniques of Dynamic Reading in your own home — at any time that is convenient for you. You can repeat exercises as often as you wish to assure that you maintain optimal reading efficiency.

SALE $44.95

Evelyn Wood *Dynamic Reader*

List $69.95

An effective and enjoyable way to improve your reading comprehension, retention and speed.

(See Page 29, 30, 54)

Home Management Sale

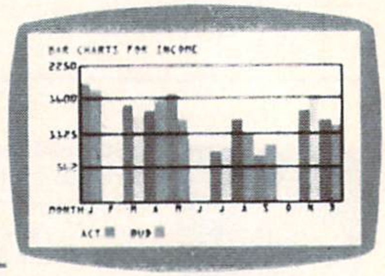




Data Manager 1

A general information storage and retrieval system — with exclusive "X-SEARCH" Feature.

List $24.95. **Sale $16.95.**

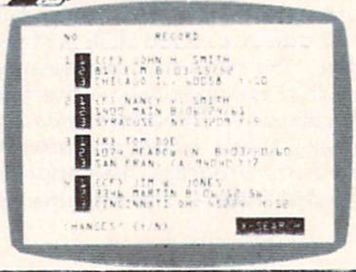




The Electronic Checkbook

Check recording, sorting and balancing system.

List $24.95. **Sale $16.95.**



The Money Manager

Home and business budget and cash flow system.

List $24.95. **Sale $16.95.**



All three Programs for only 44^{95}

Cave of the Word Wizard

For Commodore 64 Computers

An Intriguing Way to Develop Spelling Skills Using Human Speech and Arcade Action.



This state-of-the-art educational program includes 500 spoken words in 10 spelling skill levels and makes full use of the sound capabilities of your computer. The Wizard will talk to you in clear human speech. No additional hardware is needed for your computer system. (Tape/Disk)

List $39.95. **Sale $22.95.**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

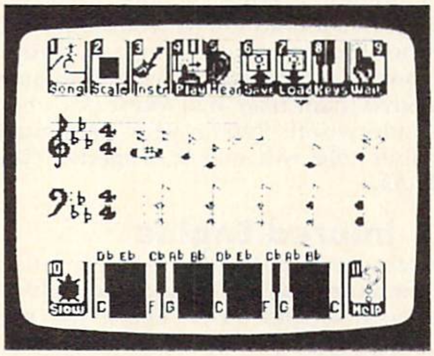
312/382-5244 to order

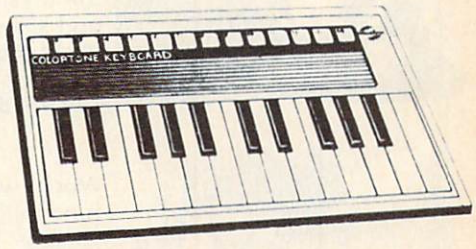
Reader Service No. 162

Commodore 64

$29.95 KEYBOARD $29.95

- Play "no-fault" music instantly
- Never hit a sour note
- Play eight different instruments
- Play along with background rhythms
- Make your own music
- Develop musical competence and confidence
- Learn notes on a music keyboard
- Develop sense of rhythm
- Begin learning music theory





ColorTone Keyboard — Now you can play and learn just like on an organ. Just point to one of the colortone's preset songs, change the musical scale you are playing in or make your Commodore 64 sound like one of eight different instruments. As you play, you'll see the notes you're hearing displayed on a musical staff then record your musical creations to listen to them again and again. List $59.95. **Sale $29.95.**

Turn Your Commodore-64 Into A Sophisticated Musical Instrument

"The Program That Gives You A Reason To Buy A Commodore-64."

New York Times.

MusiCalc 1 Synthesizer & Sequencer

MusiCalc 2 ScoreWriter

MusiCalc 3 Keyboard Maker

ScoreWriter

Combine with Musicalc 1 and a graphics printer (Super-10) to produce sheet music from your original composition. (Disk) List $39.95. **Sale $19.95.**

Synthesizer & Sequencer

This 1st step turn your Commodore-64 into a sophisticated musical instrument — a three voice synthesizer and fully interactive step sequencer play along with pre-recorded songs or develop your own and record the music you create. (Disk)

List $59.00. **Sale $29.95.**

Keyboard Maker

Turns your Commodore-64 into a musical keyboard. Comes with over 30 pre-set keyboard scales from Classical to Rock. Requires Musicalc 1. (Disk) List $39.95. **Sale $19.95.**

With Musicalc **anyone** can • Make and record sophisticated music • Print out sheet music from your creations • Turn your computer into a keyboard • **No experience necessary!**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

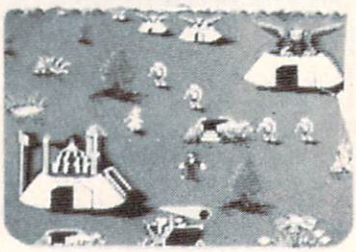
IMAGIC

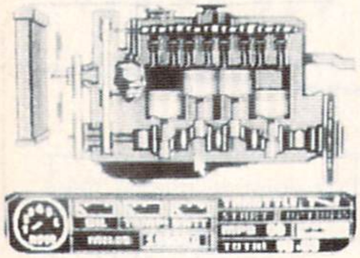
Eye Grabbing Living Graphics



The Time Machine

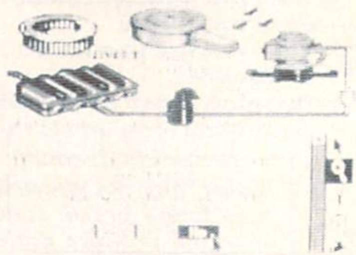
Based on H.G. Wells' science fiction classic, THE TIME MACHINE adds unique joystick action, high-resolution graphics and extensive animation to Wells' ageless prose. Travel through the treacherous time tunnel. Venture into the mysterious land of the future. Befriend the gentle Eloi and struggle to save them from evil Morlocks who lurk in the dangerous underworld. Regain your time machine at all costs — or your tale will end in tragedy! (Disk) List $39.95. **Sale $29.95.**

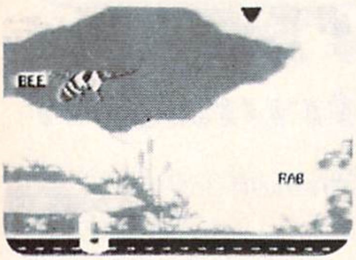




Injured Engine

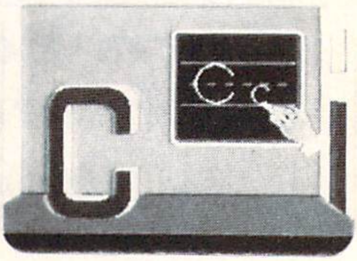
Work with an accurate cutaway representation of a functioning automobile engine. Brilliant graphics detail various engine systems and parts. Learn the names of each part and how it relates to the other parts of the engine, then, use your knowledge to diagnose computer-generated engine problems. Inspect and test at will but remember everything costs! Work against the clock to tune and repair the motor to perfect running condition. (Disk) List $39.95. **Sale $27.95.**





Speak and Seek

It talks!! Speak and Seek teaches the alphabet to children, ages 2-5. It shows children how to print letters in capital and lower cases, pronounces the letter as it is drawn and asks them to find and press the letter on the keyboard. Incorrect answers prompt a variety of helping phrases, such as "Try a little to the right" or "Try higher," and other encouraging and amusing feedback. The make a creature appear that starts with the collect letter. (Disk) List $39.95. **Sale $27.95.**



Commodore 64



Demon Attack

Winged warriors wreak havoc from above! Only your laser cannon stands between you and oblivion! Blast those bizarre demons. Your arsenal includes missiiles and sheer intestinal fortitude. Only the strong survive! (Disk) List $24.95. **Sale $16.95.**

Macbeth

Shakespeare's enthralling play of murder, greed and intrigue comes to life in this startlingly textured and vivid treatment. Armed with your wits, an impeccably detailed edition of the Scottish play and a learned mentor who questions and guides your insights and judgements, the play's the thing for you to solve! An engrossing and educational interactive adventure!. (Disk) List $39.95. **Sale $29.95.**



Commodore 64



Nova Blast

Pilot your own Novon Skysweeper above the underwater city. You must defend four cities in your quadrant from alien assault. Use your radar to detect enemy activity. Race to protect these civilized outposts in a deadly game of skill and strrategy. (Disk) List $24.95. **Sale $16.95.**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 164

NEW

Synapse

NEW

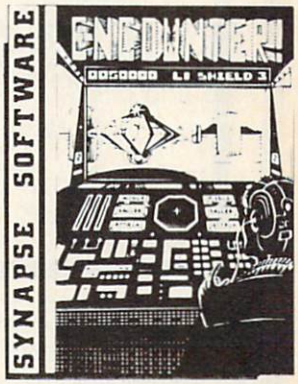
No. 1 National Rated!!

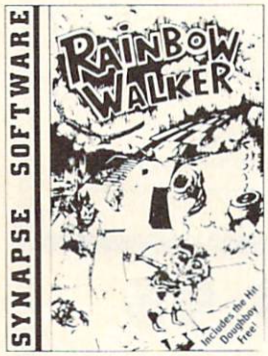
Super Action Software Sale



Blue Max 2001 — Your are Max Chatworth 9th, your mission is to penetrate enemy defenses, destroy their hover fields and finally to destroy the symbol upon which the Furxx Empire is built. The fate of the world rests in your hands. Exciting sequel to Blue Max. (Disk) List $34.95. **Sale $21.95.**

Encounter — An amazingly lifelike simulation of a futuristic tank battlefield, Encounter has only four elements — a grid made of indestructible pylons, enemy saucers, homing drones, and you. Yet these simple components create a challenge requiring total concentration and great skill. Your task — clear the grid of enemies. Success requires a thoughtful balance between sensible caution and raw courage. Keep your cool, accomplish your mission, and you enter another level — eight in all — where enemies become infinitely more clever, and so must you. Encounter is a purist's game, stripped to the essentials. This no-frills approach demands everything you've got. (Disk) List $34.95. **Sale $16.95.**





DOUBLE PLAYS

Limited Time Only

Price $19.95



DOUBLE PLAYS

Limited Time Only

Price $19.95



Rainbow Walker — Outsmart the devil and a very vicious bird, survive tornadoes and avoid the lightning bolts as the elf hops onto gray squares to create a brilliant rainbow and reach the pot of gold. You'll need razor-sharp reflexes, delicate precision with your joystick and a knack for strategy. Once you start playing, you'll find it impossible to stop.
(Includes Doughboy)

Doughboy — Don't expect DOUGHBOY to be any easier. You'll spend hours trying to outsmart the cleverest opponents while you attempt to recover the supplies that are scattered across the play field. Rocks, trees, trenches and mortars, combined with the dark of night, will surely challenge your wits and skill. Don't let your batteries in your flashlight go dead! (Disk) List $29.95. **Sale $19.95.**

New York City — You can now visit the Big Apple without leaving home, and you'll soon discover that all the stories you've heard are true! No sooner do you park your car for a visit to some of the most popular landmarks than it's stolen, and that's only the beginning. The subway is sure to make it hard to visit the Empire State Building, Central Park and Grant's Tomb. Metropolitan madness will drive you crazy — especially if you run out of gas or cash, or get hit in the crosswalk!
(Includes Air Support)

Air Support — Air Support is two games within a game providing the player a choice of an arcade game or one of strategy. Control the chopper and robots with a wide variety of commands to choose from. Select the difficulty range, terrain, number of enemy robots, airlift and bombs and begin to accomplish your mission of destroying the enemy robots. (Disk) List $29.95. **Sale $19.95.**

Quasimodo — Quasimodo is a loner who like hanging out in castle belfries. He also knows the secret hiding place where the royal jewels have been hidden. But, he is too busy stoning the soldiers ′ and swinging from bell to bell to get away from the bats. Only you can help Quasi get the jewels.
(Includes Warriors of Zypar)

Warriors of Zypar — Exciting 3D action makes Warriors of Zypar one of the most exciting two player games released this season. Placed in an arena setting, you will challenge your opponent by trying to knock him off his aero disk. Score extra points by shooting the flying ball into the mouth of the evil ZYPAR. A game that combines the barbaric adventure of the old Roman gladiators with the fast pace concept of soccer and handball all in one. Challenge your partner or your computer and play for hours. (Disk) List $29.95. **Sale $19.95.**

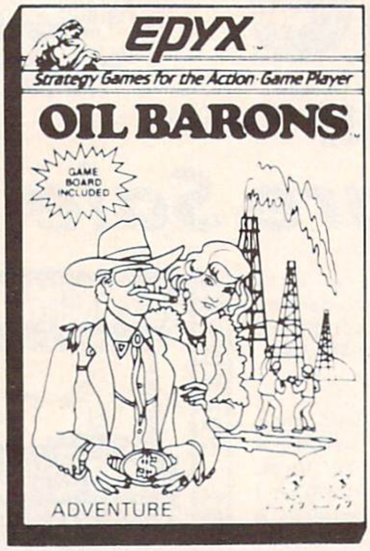
Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order



COMMODORE 64

IF YOU'RE SMART, YOU'LL GET FILTHY RICH.

FANTASTIC ADVENTURE — STRATEGY BOARD GAME!

(So good we bought out the factory)

"ANOTHER PROTECTO EXCLUSIVE"

*SALE PRICE ONLY

$19.95

*includes beautiful $8.95 leatherette dust cover for your COM-64 Computer

Factory List Price $49.95

THE BEST ADVENTURE — STRATEGY BOARD GAME IN THE U.S.A.!

Are you eager for new challenges...ready to pit your skill and luck against others in an attempt to corner the world oil market? If so, Oil Barons is your game.

Starting with 4 parcels of land and over $1 million, you search for that ever-elusive gusher. The computer does all the work, including banking and scorekeeping, freeing you to plan your strategy and make decisions that will ultimately determine your success.

Obstacles like government regulations, well fires, and hurricanes complicate matters, but the fun of increased land holdings, striking it rich, and even unloading worthless parcels on the uninitiated, more than make up for the setbacks.

FOR ALL AGES — Family, schools, businessmen, stockbrokers, oilmen etc. One to eight players, large 4 color game board, over 200 playing pieces, two sided disk and manual in a large colorful box! Keyboard controlled.

Disk Drive Cleaner $19.95

Everyone needs a disk drive doctor

(Coupon Price $16.95)

Facts

- 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless of use.
- Drives are sensitive to smoke, dust and all micro particles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

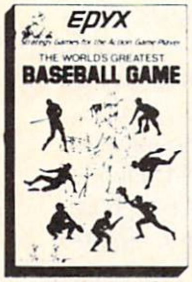
The Cheetah disk drive cleaner can be used with single or double sided 5¼" disk drives. The Cheetah is an easy to use fast method of maintaining efficient floppy diskette drive operation.
The Cheetah cleaner comes with 2 disks and is packed in a protective plastic folder to prevent contamination. List $29.95. **Sale $19.95.** ***Coupon $16.95.***

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO
We Love Our Customers
Box 550, Barrington, Illinois 60010
312/382-5244 to order

Commodore 64 EPYX™ **Commodore 64**

The Best Arcade Games Around



World's Greatest Baseball — Finally a great sports game. Realistic right down to the playing field. Control all the actions even the outfielders. Play realistic baseball all year round. (Fantastic graphics, action, and sound!) (Disk) List $34.95. **Sale $22.95.**



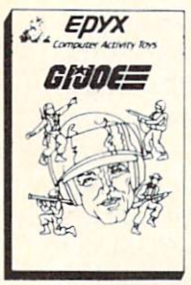
Ballblazer — Unique split-screen, 3-D graphics give you and your opponent a first person view of the field of play. You race across the playfield in your Rotofoil trying to capture the ball and fire it through the goal before your opponent. The winner is the player with the most points at the end of the timed competition. Hold onto your joystick and keep that finger on the fire button, this is the type of two player head-to-head action you've been waiting for. Two Players. (Disk) List $29.95. **Sale $21.95.**



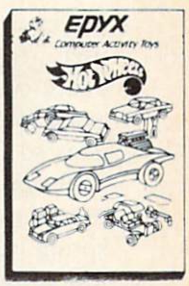
Fast Load Cartridge — Load, save and copy disks five times faster than normal. It plugs into the cartridge port of the Commodore 64 and goes to work automatically, loading disks with ease. And that's only the beginning. It can copy a single file, copy the whole disk, send disk commands, and even list directories without erasing programs stored in memory. (Cartridge) List $39.95. **Sale $24.95.**



Barbie — The biggest name in dolls. Browse through Barbie's closet full of beautiful clothing and dress her for the party. You can cut her hair or make it longer and color it or change the style. Buy new clothing at the boutique or any of the six other specialty shops, or even go to the dress shop and create Barbie's designer clothes with the help of the computer. The combinations are endless and so is the fun. One or two players. (Disk) List $39.95. **Sale $24.95.**



G.I. Joe — The best selling toy soldier is now available as an activity toy on the home computer. Select the battle situation then choose the equipment you think you will need to get the job done. Choose from a number of uniforms and weapons in your well stocked arsenal and get ready for the action. Play alone or with a friend, if you plan the right strategy you will complete the mission if not you will have to try again. One or two players. (Disk) List $39.95. **Sale $24.95.**



Hot Wheels — Now all the action and fun from playing with Hot Wheels cars is taken one step further on the home computer. Hot Wheels lets you choose your play activity from repairing cars to the destruction derby. You can actually play the activities along with a partner on the computer screen, even build and customize your own cars. A whole new way to play Hot Wheels. One or two players. (Disk) List $39.95. **Sale $24.95.**



Break Dance — Break Dance is an action game in which your dancer tries to break through a gang of break dancers descending on him. "A simon-like game" where your dancer has to keep the steps of the computer controlled dancer, and even a free dance segment where you develop your own dance routines and the computer plays them back for you to watch. Now anyone can break dance!! (Disk) List $39.95 **Sale $24.95.**



Rescue On Fractalus! — Your mission is to fly your Valkyrie Fighter through the Jaggi defenses and rescue the downed Ethercorps pilots. Sounds easy, but don't let it fool you. It's tough enough jut to navigate the mountains and canyons of Fractalus, but try doing it while destroying enemy gun emplacements or dodging suicide saucers. We supply the Long Range Scanner, Dirac Mirror Shield and Anti-Matter Bubble Torpedoes... YOU supply the skill and guts! One Player. (Disk) List $29.95. **Sale $21.95.**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 167

Disk Spinners

Continued from page 38

picts the results in no uncertain terms. A simple printer dump lets you save a permanent record of the test. The only thing the printout is missing is the provision for entering the current date. Saving a hard copy of the test will allow you to easily compare against future tests, thus periodically checking for drift in the drive alignment.

The package also includes a preformed spring wire which replaces the disk drive's hard stop. This "soft stop" will help prolong the alignment of the drive.

The only reservations we have for this product concern the accompanying documentation. The instructions, although accurate, would be of value only to someone who already knew what to expect when the drive is opened. The single photograph is too indistinct to be of any real value for anything but the coarsest details. If you have never opened the disk drive or if you are somewhat apprehensive about doing so, this is not the place to start.

Just how effective is this product? We know of at least one enterprising user group member who recoups the entire cost of the package with every alignment he performs. He even goes so far as to fabricate his own "soft stop" springs—a simple enough procedure once you find the right spring wire. As a result he has very few repeat customers. ☐

ON TRACK INDICATOR

SchuLace Enterprises
P.O. Box 771
Cascade, MD 21719
Price: $15.98

If you have ever wondered where the disk drive was when it makes all those grinding, whirring, and rattling sounds, this gadget is for you. The package consists of a very finely marked vinyl strip, some spacer material, a disk with a test program, and instructions. When the strip is properly installed, it will indicate the track to which the read/write head is positioned. Installation is straightforward, although it does involve disassembly of the drive. The manual is reasonably explicit, with several less than perfect photographs to show how it's done. Only the odd tracks have been marked off on the indicator strip as the head movement is too minute to permit additional markings. We found that the indicator was perfectly adequate for identifying all track positions.

The indicator is an invaluable aid for disk drive hackers, software copiers, or the just plain curious. It has limited value for checking head alignment. The accompanying disk includes additional operating instructions, a test program to help with the installation, and a four and a half minute, full disk copy program. ☐

REFERENCES

1. *Disk Drive Guide for the Commodore 64* by Nancy L. Wilmot, $29.98, 244 pages. Con-Cor International, 1025 Industrial Drive, Bensenville, Illinois 60106-1297. Contains ten chapters which provide a detailed tutorial on the use of the 1541 disk drive. Considerable attention is given to the matter of data files, in particular sequential and relative files. Comes complete with all of the example programs on the "friendly floppy" disk.

2. *Inside Commodore DOS* by Richard Immers and Gerald G. Neufeld, $19.95, Datamost, 20660 Nordhoff Street, Chatsworth, CA 91311-6152, 818-709-1202. The definitive hacker's handbook on the innermost workings of the 1541 DOS. A must-have for all advanced disk users who wish to get at the DOS from a programming standpoint.

3. *1541 Single Drive Floppy Disk Maintenance Manual* by Michael G. Peltier. $39.95. Peltier Industries, Inc., 735 N. Doris, Wichita, KS 67212. 316-945-9266. The 1541 hardware hacker's Bible. Meticulously and lavishly illustrated. Provides detailed descriptions of the mechanical and electrical workings of the 1541 disk drive. Includes instructions on the care, feeding, and alignment of the 1541. ☐

Reviewed next issue:
Kwik-Load!
Fast Load
1541 Flash

REVIEWS

Continued from page 28

game-playing stage, engaging and coherent instructions would have sounded much better.

If you have the tenacity to muddle through, however, you'll find much enjoyment in accompanying Indy on his further adventures. A fellow reviewer once wrote that a good game was one he'd return to play after the review was written. For me, *Indiana Jones* is one of those games.

Mindscape, Inc., 3444 Dundee Road, Northbrook, IL 60062 (phone: 312-480-7667).—***Robert J. Sodaro***

HOMEPAK

Batteries Included
Commodore 64
Disk; $49.95

It sounds like a great deal. $49.95 for a word processor, information manager, and terminal program. But don't expect more than you pay for. As an integrated software system, *HomePak* has its faults. However, depending on how much you intend on using the individual programs, it may be exactly what you need.

The word processor, called *HomeText,* is quite nice. It allows a sophisticated mail merge operation with the database, *HomeFind*, with a limit of 80 characters on the information merged. *HomeText* uses the normal cursor keys for general editing, only requiring the generation of such nasties as CTRL-Q and COMMODORE-S for specialized functions (such as moving the cursor to home and changing screen color).

I find that when it is necessary to memorize lists of keystrokes to use advanced word processing operations, I don't. I just survive using the obvious. *HomeText* avoids this problem somewhat by minimizing coded keystrokes and supplying a reference guide which lists all the control keys and their functions.

HomeText supports block manipulation, search and replace, and file merging. The file merging function is one way to bypass a file size limit. *HomeText* also had a print preview operation. After the user sets up the printer format for a specific docu-

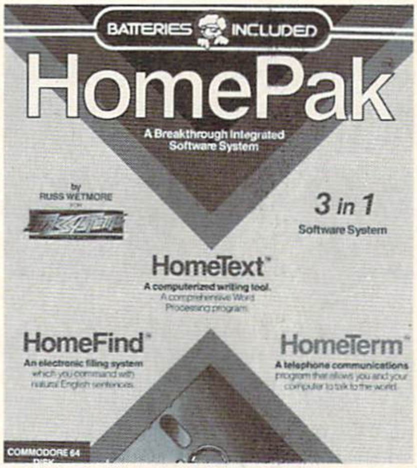
ment, *HomeText* can display a print preview screen in high-resolution graphics. The print preview shows extended and boldface words in yellow, underlined words with a green line, and normal words in black. The page is formatted as it will appear when printed. I found this function extremely useful, especially after having printed documents over and over in the past, trying to gently persuade a word processor to print a document the way I needed.

After looking at *HomeText, HomeFind* was very disappointing. Anyone who needs to use the database for even a mildly sophisticated operation will be frustrated and confused. *HomeFile* is a "natural language" database, which means that there is very little formalized information structure. Data is input in a subject-tag-object format such as "John's phone's 555-1234" or "Susan's birthday's May 15, 1951." If specific information is to be recalled, one would input "What's Susan's birthday?" Then, the subject-tag-object is recalled and output. It is possible to do limited searches by inputting "Who's Susan?", in which case all data about Susan would be output. It seems a silly way to construct a database.

It might be easier for a novice to use natural language, but even novices eventually need to perform mathematical operations within a database, or other more advanced functions. The traditional field-record-file database structure can be confusing to inexperienced users, but once the organization becomes intuitive, there is more freedom within the logical system. After all, it is always necessary to learn before using new utility programs. Under a natural language system, complicated operations are complicated and, in some cases, impossible.

Another major fault of *HomeFind* is its lack of a report generator. This was justified in the documentation by the statement that "...most home users don't need a fancy printout of their data." All that is possible is to echo data put on the screen to the printer. Well, I want to be able to decide what I need for myself. *HomeFind* also supplies a quick reference guide, for whatever it is worth.

HomeTerm, the final program in the package, is equal in quality to *HomeText*. It is wonderful. The manual initially gives some information on the basic technicalities of telecommunications. *HomeTerm* supports transmission of ASCII, C-64 ASCII, VIDTEX, and X modem format data. There are two *HomeTerm* screens, the interactive mode screen and the functions menu screen. There is an edit window in the interactive mode screen which allows control of information before it is transmitted. It is also possible to store incoming information in a data capture buffer which can be saved to disk file or dumped to a printer or the screen.



A trio of integrated home utilities.
READER SERVICE NO. 143

There is also a disk files functions menu within *HomeTerm* which allows you to perform DOS operations without leaving the terminal mode, or even going offline. A timer can be set to real time or used to keep track of time spent online. But the best special feature is the define macro command (macro) sequence.

A macro is created when a string of commands is shortened to a single keystroke. Thus, a sequence such as the list of commands used to log on to a BBS can be defined as a macro and encoded into a single keystroke. *HomeTerm* allows definition of ten macros and provides for them to be saved in a special macro configuration file. In fact, it is possible to save all of the user-defined modem configuration parameters to a file and recall them when entering *HomeTerm*.

The last part of *HomePak* is one which is not announced in bright red letters on the cover of the box. It is *HomeTrans*, which converts one type of file to another. This program is what "integrates" the software package. Three basic file types are generated by *HomePak:* ASCII, *HomeTerm* PET ASCII, and special *HomePak* files. These types of files are *not* compatible with those of other parts of *HomePak*. In order to use one type of file in another part of the program, it is necessary to through *HomeTrans*. Because of the speed (or lack thereof) of the 1541 disk drive, converting files is more than a small inconvenience. It is more of a major annoyance.

All in all, *HomePak* has some well-designed and prepared programs. *HomeText* and *HomeTerm* are easy to learn, easy to use and very well designed. *HomeFind*, on the other hand, is a total mess. But, for the price, getting good word processor and terminal programs seems just about fair.

Batteries Included, 186 Queen St. West, Toronto, Ontario, M5V 1Z1 Canada (phone: 416-596-1405; US number 714-979-0920).

–Rachel Schleimer



"Testing, 0001, 0010, 0011..."

SHIP TO SHORE

Continued from page 60

is all it takes to prevent problems, but not all programmers know how to do it. Being associated with others who have experienced the problems and solved them can really help.

SYSOP associations are only one way of combatting the problems that BBS owners run into, but more and more of them are finding that togetherness can have an impact. Of course, the people who really benefit are the users of the boards. They can spend more time on the board, have to wait less time to get on, and know that their privacy is secure. When properly run, they can be a great boon to all users.

With recent legal developments, SYSOPs have even more reason to stick together. More than a dozen states have enacted stiff laws designed to counteract computer criminals. In at least one case, this seems to be back- firing.

In LA, one SYSOP, Tom Tsimpidis, had his computer equipment and most of his diskettes confiscated when Pacific Bell Telephone informed the police that illegal telephone access codes had appeared on his board, MOG-UR. Two weeks later the equipment was returned and the lawyers involved seemed to think the subject was closed. SYSOPs nationwide breathed a sigh of relief, a bit prematurely it now seems.

At this writing, Tsimpidis is scheduled to appear in court for arraignment on unspecified charges. SYSOPs from several states have offered to appear as character witnesses, and several lawyers have volunteered research time to help get the charges dismissed. Ongoing discussions on the LAW SIG and Writer's and Editor's SIG of CompuServe tend to provide overwhelming support for Tsimpidis, but they also seek ways to prevent similar occurrences.

Some SYSOP have started a policy of only posting messages after they've been approved by one or more of the BBS operators. In some cases, trusted members are also granted approval rights. This system tends to slow down the message traffic, however. For quick response boards, this could be fatal. A Chicago group, the National Association of Bulletin Board System Operators (NABBSO), started investigating the legal precedents for SYSOP responsibility when Tsimpidis' problem initially came to light. Although everyone assumed Tsimpidis would have no need for the resulting information, the legalities search continued because honest SYSOPs thought they needed to know their legal responsibility. Could they be held liable for illegal messages left by others on their board, even though they did not approve the message? Does even one illegal message for only a day prove intent to defraud? One message for three hours? Three messages for one hour each on different occasions? Where is the line? NABBSO and other SYSOP organizations need to know, so that they can protect their members.

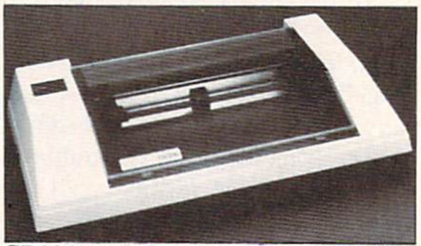
There are many illegal BBS's that honest SYSOPs would like to see taken down. In this case, though, the SYSOP is known for running a legitimate BBS. Many SYSOPs fear that a bad precedent could be set if this case is successfully prosecuted. If a BBS operator is responsible for everything that appears on his board, regardless of the source, privately run BBS's will probably disappear. 24 hour monitoring would be extremely difficult, and part-time BBS's aren't often successful.

SYSOP organizations have been formed to help prevent illegal activities on privately operated systems. By cooperating, SYSOPs have started to protect themselves from unethical users. Powerful laws were enacted to help stem the flow of criminal computerists and honest SYSOPs welcomed them. Now, these laws may be backfiring to threaten the very people they were designed to protect. The legal system may be the factor that finally overpowers free electronic bulletin boards. ☐

SCUTTLEBUTT

Continued from page 16

EnTech has made its *Studio 64* music synthesizer program compatible with Sequential Circuits' Music Mate keyboard. Commodore keyboard and Sequential Circuits keyboard versions will be sold on the same disk for $39.95; current *Studio 64* owners can get the update by sending $10 and their old disk to EnTech Software, P.O. Box 185, Sun Valley, CA 91353 (phone: 818-768-6646).



CR220: printhead hits rotating roller.
READER SERVICE NO. 170

PRINTER OUTPUT

Sakata's SP-1000, operating at 100 cps, has high throughput features with bidirectional logic-seeking (66 lines per minute), replaceable print head, and adjustable tractor and friction paper feed. Price: $369.00.

The 120 cps SP-1200 serial printer offers logic-seeking or incremental printing with high-response stepping motor, fixed and proportional print modes, and an eight-language international character font. Price: $399.00.

Sakata U.S.A. Corporation, 651 Bonnie Lane, Elk Grove Village, IL 60007 (phone: 312-593-3211).

The Riteman C+ with built-in Commodore interface can be plugged directly into the computer's serial port. Front-loading design allows paper to snap into the front tractor sprockets without alignment problems; the built-in stand lets you store paper under the printer. True descenders and 82 Commodore graphic characters are produced.

Inforunner Corporation, Airport Business Center, 431 N. Oak Street, Inglewood, CA 90302 (phone: 213-453-6688).

Two printers from Epson America:

Unlike in conventional dot matrix printers, where multiple pins strike a stationary roller, the Comrex CR220's Uni-Hammer is a single printhead that strikes a rotating roller. The 50 cps printer features a built-in Commodore interface. Price is $199.

HomeWriter 10 offers draft (100 cps) and near letter quality (16 cps) modes, and (via Epson's SelecType feature) a variety of typestyles. The $269 printer will require a $60 Printer Interface Cartridge to interface with your C-64. Other features are bi/unidirectional printing, 1K buffer, and optional tractor ($39.95) and cut sheet ($99.95) feeders.

Epson America, 2780 Lomita Blvd., Torrance, CA 90505 (phone: 213-539-9140).

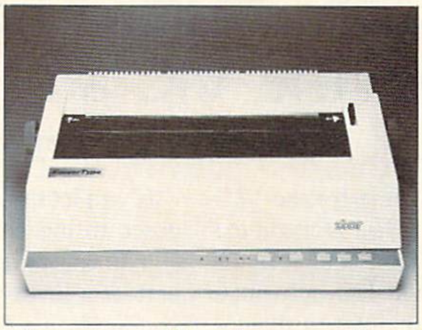
Star Micronics has introduced three new printers, in 10 and 15 inch versions (the wider models also featuring a 16K buffer). The SG-10/15, SD-10/15, and SR-10/15 operate at 120, 160, and 200 cps respectively, with, we're told, 20% faster throughput. All feature near letter quality and draft modes, standard friction and tractor feed (the SR's add single sheet), and hi-res bit image graphics.

Also from Star comes the Powertype daisywheel printer, featuring 18 cps, bidirectional, logic-seeking printing in 110, 132, or 165 columns. SG-10: $299; SG-15: $499; SD-10: $449; SD-15: $599; SR-10: $649; SR-15: $799; Powertype: $499.

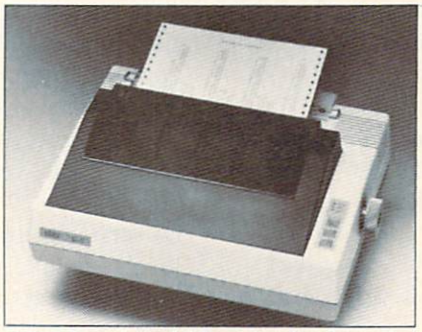
Star Micronics, Inc., 200 Park Avenue, New York, NY 10166 (phone: 212-986-6770).

The Juki 6300 offers 40 cps daisy wheel printing for $995. Included is a 3K buffer (expandable to 15K) and a wide range of special print features.

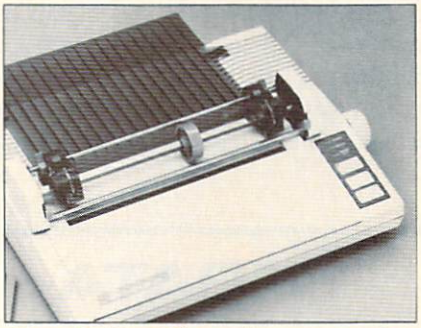
Juki Office Machine Corporation, 1261 Wiley Rd., Suite B, Schaumburg, IL 60195 (phone: 312-843-3322).



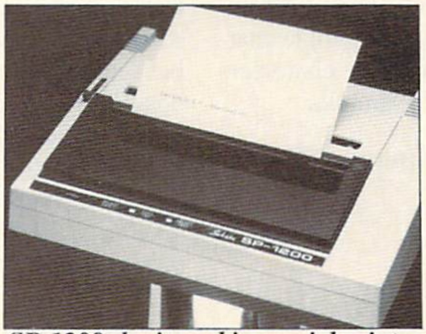
Powertype: 18cps letter quality.
READER SERVICE NO. 171



SD-10: prints 120cps in draft mode.
READER SERVICE NO. 172



HomeWriter 10: various typestyles.
READER SERVICE NO. 173



SP-1200: logic-seeking serial printer.
READER SERVICE NO. 174

TERMINALLY YOURS

What's new, and of interest to Commodore owners, in the world of telecommunications:

Orbyte's *Hotline Communications* program accesses most databases, uploads and downloads keyed, disk, and gathered data, and reads information from a database into memory and displays it onscreen for scanning, editing, or saving. Price: $79.95.

Orbyte Software, P.O. Box 948, Waterbury, CT 06720 (phone: in CT 203-621-9361; rest of US 1-800-253-2600).

CompuServe has expanded its online sports services to include facts and figures on college and pro sports, updated as often as five times hourly.

CompuServe, 5000 Arlington Centre Blvd., P.O. Box 20212, Columbus, OH 43220 (phone: 312-527-5100).

ECHO (Electronics Communications for the Home and Office) is expanding its services to include an optional hard copy of all electronic mail sent through the system, allowing subscribers to send E-mail to persons who don't own computers. An ECHO-Express Letter of up to four printed pages will cost $1.75.

ECHO, 4739 Alla Road, Marina del Rey, CA 90091 (phone: 213-823-8415).

The Association of Electronic Cottagers (AEC) has been formed to provide business consultation and other services to persons who work out of their homes with computers. Free information is available from AEC, 677 Canyon Crest Drive, Sierra Madre, CA 91024 (phone: 818-355-0800; CompuServe no. 76703, 242).

XNET's Adult Electronic Mail Computer Network features romance-oriented personal ads. Special rates are available for new subscribers.

XNET Computer Services, P.O. Box 2365, Halesite, NY 11743 (voice phone: 516-549-0811; computer: 516-549-0845).

Answers Online: Your Guide to Informational Data Bases describes the majors and the most cost-effective means of searching them. $16.95 from Osborne/McGraw-Hill, 2600 Tenth St., Berkeley, CA 94710 (phone: 415-548-2805).

The *Data Base Directory* compiled by the non-profit American Society for Information Science lists and details over 1800 services in North America. The 608-page volume is available for $120 from Knowledge Industry Publications, Inc., 701 Westchester Ave., White Plains, NY 10604 (phone: 914-328-9157).

HOP AROUND
for the VIC and 64
By Kevin Dewey

In this fast-action arcade style game for the VIC and 64, a demon has cornered Sweet Doris at the top of a five-story construction. She cannot run—the demon has cast a spell on her. Her only hope is that you can scale the construction and defeat the demon who is inching his way towards her.

The demon knows you're there. In fact, he's toying with you, dropping bombs in your direction. His bombs are making holes in the construction, which he thinks will be to his advantage. He's wrong.

He doesn't realize that you have awesome vertical jumping power. You can jump through the holes and make your way to the top, where you can banish the demon by jumping on his head as he hovers over a hole in the construction.

Here are some fundamentals of the gameplay:

If you walk off the right side of the screen, you'll appear on the other side, falling.

If you walk off the left side, you'll appear on the other side on the same level, provided that there isn't a blank space where you appear—if there is, you'll fall through it.

If you fall off the bottom of the screen, or fall too great a distance within the construction (two levels in some cases, three in others) the game is over.

If you jump or run into a bomb or the demon, the game is—you guessed it—over.

If you let the demon reach Sweet Doris, he'll take her down to the demon world.

You can jump as high as you want: if you jump into part of a level you'll bounce to the right and fall until you land on a level. You cannot jump right or left, or over gaps. You never bounce left when you jump into part of a level—this would make the game too easy.

When you win the game and the demon finishes his wild plummet down to the demon world, a text screen appears telling you how many jumps it took you to win. You can try to win in a low amount of jumps or a high amount: both ways are challenging.

In the VIC version you control your man by pressing "J" to jump, ":" to move left, and ";" to move right. The 64 version is joystick-controlled. The 64 version has five levels of skill, selected via the f1 button when the title screen appears. (Note: when the winning or losing text page prompts you with "HIT ANY KEY," press f1. That's the only way you can keep it from zipping by you, and change levels.)

When you begin to play, all the characters on the screen are black. You can change this color to any of your VIC or 64's available colors by pressing f1 when you see the winning or losing text page. There are even a few weird multicolors you can choose. □

SEE PROGRAM LISTING ON PAGE 97

ERRATA

Due to a flaw (since corrected) in the program we use to generate the listings that appear in the magazine, several errors occurred in *Ultra Mail* and *Alice in Adventureland* (January '85 *Ahoy!*). In strings of multiple characters, the numerical portion of all multiple characters after the first increased by one. For example,

```
PRINT"[5"[DOWN]"][15"[RIGHT]"]"
```

should read

```
PRINT"[5"[DOWN]"][14"[RIGHT]"]"
```

Errors like this occurred in the following lines:

Ultra Mail: 2, 2100, 4015, 4045, 6000, 6100-6800, 7020, 7030, 12065

Alice in Adventureland: 224, 1300, 1700, 1710, 2000, 2020, 2030, 2050, 2100, 2500, 2503, 2590

Curiouser and Curiouser—*Alice* had another apparent error crop up. Several readers claimed they received a BAD SUBSCRIPT or ILLEGAL QUANTITY error in line 1100. Our investigation of the problem bore out the theory that there's an exception to every rule: by transposing certain numbers in the DATA statements of *Alice*, it is possible to type information in incorrectly but receive the correct *Bug Repellent* code. These transposition errors can be caught only by manually proofreading lines 9610-9806.

Also, Donald Graham wrote in to tell us of an omission of his in his *Exposé of the VIC & 64 Operating System* (also Jan. '85). Line 500 as printed is for VIC users only. C-64 users should substitute the following:

```
500 IF PEEK(197) <> 1 THEN 500
```

Thanks also to Heike Hamacher of Tulsa, Oklahoma for picking up the error, figuring out the correction, and sending it to us.

Our apologies for any problems these errors may have caused. Remember that corrections to *Ahoy!* programs and articles are posted on our bulletin board (212-564-7727) as soon as they are spotted. □

COMMODORE
-USER WRITTEN SOFTWARE-

Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

COMMODORE 64™

64 collections # 1 thru 12
25 + programs per collection-Tape or Disk $10.00 each

COMMODORE 64 COLLECTION #1
3D Tic Tac Toe ● 64 Memory Chart ● 64 Renumber ● Baseball 64 ● Black Box ● Checkbook ● Chess for Two ● Concentration ● Connect Four ● Dr. Living ● Dragon Pix ● Entertainer ● Fugue ● Gun Fighter ● Hangman ● Ink Blot ● Joystick Test 64 ● Jumbo Jet Lander ● Labyrinth (J) ● Lander ● Machine Draw ● MeSprite ● Old Shell Game ● Petals ● Program Lister ● Scenarios ● Sprite Ed. Disk ● Sprite Ed. Tape ● Stock Broker ● Trek 64 ● World Time

COMMODORE 64 COLLECTION #2
Hires Disk ● Hires Tape ● Cave Man ● Man ● Pattern ● Sattlite ● Snail ● American Flag ● Baby Care ● Bar Graph ● City Paterns ● Epidemic ● Flight ● Forest Walk ● Geiger Counter ● Life Expectancy ● Master Guess ● Monopole 64 ● Nuke 64 ● O Hare's Adv. #1 ● Othello for 2 ● Patterns ● Pharoah's Needle ● Poker ● Reg'd Supply ● Supermon 64 ● Temp. Conv.

COMMODORE 64 COLLECTION #3
GAMES — Biorythmn 1.5 ● Checkers ● Clew-So ● Dam Busters ● Hi-Q ● Mille Bourns In ● Mille Bourns PR ● O Hare's Adv. #2 ● Pig ● Wrap Trap
UTILITIES — Base ● Bio Printer ● Bond Yield ● Calendar ● Dissambler ● FICA Tax ● Food Prices ● Heating Cost ● Intermod ● J/20 Morse R/T ● Menu ● Money Exchange ● PCBN Printer ● Portfolio ● Property Eval ● Timer ● Typing Aid ● Typing Test ● View Disk ● Weight Watcher
EDUCATIONAL — Chemical Drill ● Coll. Like Terms ● Cryptograms ● ESP Test ● French Sentence ● French Verbs ● Geography ● Grades ● HS Layout ● Integer TTT ● Latin Tutor ● Math Drill ● Metric Quiz

COMMODORE 64 COLLECTION #4
GAMES — Bttleship ● Bomardier ● Crops ● General ● Key Move ● Letter Square ● Money Mad ● One Checker ● Osero ● Spades Inst ● Spades Prog ● Stoneville Manor
UTILITIES — Apartments ● Banner ● Big Letters ● Car Cost ● Cash Flow ● Data R/W ● Directory ● Fast Sort ● Global ● Grades ● Interest Rate ● Print Format ● Printer Helper ● Servings ● Sunrise/Sunset ● Weatherman
EDUCATIONAL — Density ● Fractions ● Grammer ● Integer Math ● Long Division ● Mechanics ● Powers & Roots ● Straight Line ● Whole Numbers ● Word Man

COMMODORE 64 COLLECTION #5
GAMES — Alien Capture ● Auto Rally ● Dominoes ● Explore Lost Gold Adv. ● Niche ● Owari ● superman ● Zone X
UTILITIES — Accounting Test ● Annuity ● Average Growth Building Costs ● Buy/Sell ● Change Disk Name Cross Word ● Data Search ● Elec. Use ● Finance ● Growth Irr. Cash Flow ● Load Calc. ● Real Estate cost ● Rent or Buy
EDUCATIONAL – A or An ● Arrow Facts ● Big Math ● Element Chart Eliza ● Europe ● Hang Math ● Linear Prog. ● Loops Marblestat ● Math Facts ● Number Match ● World Quiz

COMMODORE 64 COLLECTION #6
GAMES—Action Hockey ● Backgammon ● Civil War ● Dixie 64 ● Have a Nice Day ● Mission to Mars ● Octopus 1 ● Race Saucer Invasion ● UTILITIES—64 Label ● Alpha Index Dates ● Decision Maker ● Disassembler ● Event Calendar Find Nom. Int. ● Income Tax Adv ● Investment F.V. Music Driver IN ● Music Driver P1 ● Music Driver P2 Music Driver P3 ● Music Driver P4 ● Pareto ● Roman Numerals ● Seq. Append ● Term 64 ● Voice 3 Music
EDUCATIONAL—A Fable ● Curve Fit ● Finance Calc. ● Foil Inst ● Foil Pract ● Heat Solver ● Intergration ● Percent Drill Projectile ● Titrator 64 ● Weigh

COMMODORE 64 COLLECTION #7
GAMES—Alpha Base 64 ● Blackjack/21 ● Diffuser Doctor ● Las Vegas Slots ● Pet Bomber ● Pictures Randomizer ● Speed Racer ● UTILITIES—Astro-bioprinter C/S Retirement ● Days Between Dat ● Disk Ease ● Disk Utility ● Elect. Formulas ● Home AC G Input ● Home AC G Sum r ● Lister V64 ● Load Address ● Mortgage ● Ohms Law Quadratic Equ ● Sound Master
EDUCATIONAL—A1 Arrows ● Capitals ● Factor Race Flash Words ● Histograms ● Matrix-Equation ● Phoneme Words ● Population ● Reading Comp. ● States Quiz Statistics ● Triangle Solve

COMMODORE 64 COLLECTION #8
GAMES—Card Guess ● Chain-Reaction ● Doors ● Football Obstacle Course ● Pyramid ● Roulette ● Starwar ● Tiles
UTILITIES—Astro-Biorhythm ● Cross-Ref 64 ● Energy Saver Investment Calc. ● Marks-Disk ● Merger ● Metric Convers64 ● Sort Demo ● Tiny Forth 64 ● Tiny Forth Inst Typewriter Disk ● Typewriter Tape ● Utility Menu
EDUCATIONAL—African Caps ● Asian Caps ● Chemists Factors ● FN Machine ● Function Plotter ● History Quiz Multiplot ● Pizza ● Prime Numbers ● Signed Numbers Tax Collector ● Vector Solve

COMMODORE 64 COLLECTION #9
Menu Loader ● City Manager ● Gladiator ● HS of 7 Gables ● King Tut Sorcerers Castle ● ● The Valley ● Trip to Atlantis Wizard's Calstle ● Westward Ho!

COMMODORE 64 COLLECTION # 10
64 Yahtzee ● Address List ● Africa Safari ● Animal Math 1 Animal Math 2 ● Animal Math 3 ● Animal Math 4 ● Blockmod4 ● Bowling C64 Color Test ● C64 Contest Log ● Castlemaze Adv'r ● computer Sketch Converge ● Craps ● Dancing Idiot ● Dial 3.2 ● Disk of Fortune ● Disk Util. Dixie ● Doctor Dementia ● Dpt 3.8 ● End Subroutine ● Flash Cards Hangperson ● Hex 5.0 ● Investment File ● Mod Index ● One Liners Quiz-Jesus & Law ● Quiz-Jesus Seid ● Recipes ● Russian Roulette Scramble ● Wordscramble

COMMODORE 64 COLLECTION #11
A Sequence ● Alphie ● Bridge ● C&C of Afr.64 ● Camel II ● Christmas Card ● Color Combos 64 ● Dungon Master ● Flight Simulator Guadish Eye ● Graph Builder ● Graph I ● Graph II ● Hamurabbi Invest.File/Prt. ● Kiddie Code ● Lemonade Stand ● Line/Prog Linker Marquee ● ML Loader ● ML Saver ● Mult'n Tables ● Music Lesson Packadot ● Receipt Master ● Rotate ● Ratate 90 ● Thirsty Nellan Wheel of Gold ● Word Order

COMMODORE 64 COLLECTION #12
Add.Load ● Address Book ● Arrow ● Auld Lang Syne ● Colormon Loader Colormon ● Computer Song ● Creation ● Daily Compound'g ● Decision Dir.Print ● Dos 5.1 Commands ● Dos 5.1 Sub ● Dos Control Down The Chimney ● Drip Simulator ● Hang-Math ● Jstick Doodle Kepler.Elements ● Lonrep ● Math Menu ● Menu ● Metric Mortgage Printer ● Mr. Dig ● Octopus Maze ● Othello ● Paycheck Verify Peek 197 ● Ro Skier ● Prime Numbers ● Print Using Sub ● Snoopy Snoopy Calendar ● Solitare ● Std Deviation ● Super Star Trek ● Swerve T.V. Satellites ● Tictacarith ● Typing Test ● Video Vandal

P.D.I. PROGRAM MANUAL - $10.00

The P.D.I. Program Manual will be the same size as the Commodore manuals, and each program on volumes 1 thru 12 of our Commodore 64 or Vic-20 Software Collections is given a review. If a program requires additional hardware, such as memery, joysticks, disk or printers, we make note of it. If a program requires additional instructions, other than those in the program you will find them here.

* Plus *

Helping Hints' (disk chatter and seq. file Dynamic Loaders) and more funs (The new Pet Rock and Computer Widow).

DINSET™: Reset Switch
Works on Vic 20 or Commodore 64 — $5.00

VIC 20™
Vic 20 collections #1 thru 12
50 + programs per collection-Tape or Disk $10.00 each

SERIAL CABLES
10 FT.—$10.00
15FT.—$15.00

PET®/CBM® Software Available

P.D.I. PROGRAM MANUAL - $10.00

GA Now Available For Vic Owners, Too

Disk Only $10.00

Now GA available for the Vic, Pet & 64 owners. The original Cavern Adventure on GA, has been modified for those Vic 20 owners that are Disk Based and have 16K+memory.

HAMFEST COLLECTION HS
Tape or Disk Only $10.00
COMMODORE 64—Air Inductor ● DB Calc. 64 ● Dupe 64 ● E. Brs 64 Elect. Formulas ● Ham Formulas 64 ● Intermod ● J/20 Morse R/T L Pass Filter 64 ● Ohms Law ● PI Network ● Desig ● Quad Antenna 64 Reg'd Supply ● Res. Freq 64 ● Rtty-C64 Inst ● Rtty-C64 Prog T.V. Satellites ● Voltage Div. 64 ● Yagi 64
VIC 20—Air Inductor ● Bob's Beam ● Call Search ● Code Practice Coverage Predict ● CW Keyboard ● Decibel Calc. ● Dupe ● E-Brg Electronics 1 ● Ham Contest Log ● Ham Formulas ● J/20 Hex P J/20 Hex T ● J/20 Recv/Trns ● Lo Pass Filter ● QSO Directory ● QSO Vic Quad Antenna ● Reson. Freq. Formu ● Rtty-Vic 20 ● Rtty 16k-Inst Rtty-16K-Prog ● Sunrise/Sunset ● Voltage Div. ● Yagi

LOC-LITE™ Operation Status Indicator
Assembled and Tested $20.00
Kit w/inst. $15.00 Board w/inst. $7.50

Prices include U.S. shipping and handling only.
CHECK, MONEY ORDERS, VISA and MASTERCARD accepted. NO C.O.D.'s
Write For A Free Flyer Or Send 60¢ In Coin Or Stamps For A Complete Catalog.

'PUBLIC DOMAIN™, INC.

5025 S. Rangeline Rd., West Milton, OH 45383
10:00 a.m.-5:00 p.m. EST—Mon. thru Fri.
Dept. Ahoy!
(513) 698-5638 or (513) 339-1725

VIC 20™, CBM® and Commodore 64™ are Trademarks of Commodore Electronics Ltd. PET® is a Registered Trademark of Commodore Business Machines, Inc.

Reader Service No. 116

HOME CONTROL

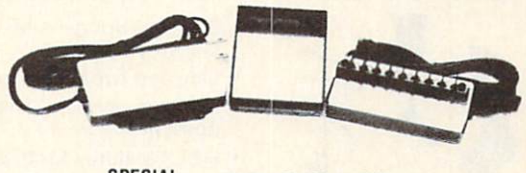
HOME SECURITY AND ENERGY MANAGEMENT PRODUCTS

VIController. Wireless home control using BSR or Leviton receiver modules. Software included. $69.95

COMclock/AB Autoloads and runs program at power-up. Battery backed up clock. (C-64 only) $69.95

COMsense. Provides 4 open/close and 2 analog inputs. Great for security $69.95

SUPER SCHEDULE PLUS Control software utilizing above three products. (C-64 only) $19.95



SPECIAL PACKAGE PRICE **$199.95**

SPEECH

COMVOICE! THE EASY TO USE SPEECH SYNTHESIZER FOR THE C-64

SPEECH IS AS EASY AS 1-2-3!
1) Plug COMvoice in.
2) Turn computer on.
3) Type SPEAK "HELLO, HOW ARE YOU"

AS EASY TO USE AS A PRINT STATEMENT!

$99.95



W/EXTERNAL SPEAKER $139.95

EDUCATIONAL DEMO & AARDVARK ADVENTURE SOFTWARE AVAILABLE

Reader Service No. 117

GENESIS COMPUTER CORPORATION • P.O. BOX 152 • HELLERTOWN, PA 18055

DEALER INQUIRIES INVITED • **(215) 861-0850** • MC & VISA Accepted

INTERFACING UPDATE—
CARDCO'S OuiG

We received a prototype of Cardco's latest entry into the Commodore printer interfacing arena just a shade too late to include it with the interface reports in our last issue. The Card/?OuiG (pronounced "card print wee gee") will be replacing the Card/?+G as Cardco's entry for a total 1525 emulation interface.

The interface sports the same features as the +G (reported on in the March '84 and '85 issues), with some significant improvements in its operating system. By doubling the size of the onboard ROM, the OuiG becomes the second interface of our test group to provide total emulation of the 1525 with all three of our test printers (the Micrografix MW350 and its clone, the Easy Print with Graphics, were the first). At a selling price of $69.95, the OuiG is also the lowest cost interface to provide this capability.

The new operating system includes three versions of the Commodore special graphic characters, thus allowing the interface to generate optimum print, with the proper aspect ratio, for all printer character matrices spanning six, seven, and eight dot positions.

The result is that proper aspect ratio will now be obtained by the Epson and Star Micronics brands of printers as well as all the lookalikes. The QuiG still lacks any significant onboard RAM buffer. This means that the operating speed for the graphic-related operations in emulate mode will be somewhat slower than possible with the state of the art. For example, the single and double size high resolution bit map dumps created with the *DOODLE!* graphics package require three minutes and fifteen seconds and thirteen minutes and twenty seconds respectively when done in 1525 emulation mode on a Gemini 10X printer. This is still a significant improvement over the performance obtained with the +G. As of this writing (January 1985) the QuiG prototype was configured only for the C Itoh and Epson type printers. The Okidata configuration will be incorporated in the final version, which should be ready by the time you read this.

The improved performance was obtained by careful use of the 64 bytes of RAM included with its 8035A microprocessor, allowing the interface to look ahead twenty bytes when doing a graphics dump with a Gemini 10X or Epson printer. This reduces the number of printhead oscillations to only twenty for a 400 pixel graphics row.

The OuiG comes in Cardco's new compact package with the entire works attached to the Centronics printer connector. The need for any flat ribbon cable has been eliminated. Five volt power for the interface is obtained from the computer's cassette port. The OuiG will of course be covered under the terms of Cardco's lifetime warranty.

For information contact Cardco, Inc., 300 S. Topeka, Wichita, KS 67202 (phone: 316-267-6525). ☐

—Morton Kevelson

COMMODARES

Continued from page 63

```
240 GOSUB 300
250 X2=6*N+7
260 GOSUB 300
270 NEXT N
280 GOTO 380
300 X4=INT(X1/X2)+X2
310 X4=INT(X1/X2)*X2
320 IF X4<>X1 THEN 370
330 PRINT X2;
340 X1=X1/X2
350 IF X1=1 THEN 390
360 GOTO 310
370 RETURN
380 PRINT X1
390 PRINT:PRINT TAB(11)CHR$(18)"PRESS AN
Y KEY"CHR$(146)
400 GET K$ : IF K$="" THEN 400
410 RUN
```

Milton Powell (Portsmouth, NH) has sent a discussion of some of his discoveries regarding the creation of protected disk files. Mr. Powell's method saves a program file in such a way that it may not be readily loaded from the disk by anyone not knowing the secret. Here is a summary of the steps to save such a program with the filename "123":

1. Type in or load the program normally
2. Type SAVE"" (no spaces, no return)
3. Backspace with INST/DEL one space
4. Press CTRL-9 [RVS-ON]
5. Type T
6. Press CTRL-0 [RVS-OFF]
7. Type the filename 123
8. Press CTRL-9 [RVS-ON]
9. Type TTTTTTTT [8 RVS T's]
10. Press CTRL-0 [RVS-OFF]
11. Type ",8 [quote key followed by ,8]
12. Press RETURN

If you look at the directory, all you see is a quotation mark and the PRG file type. To load the program, repeat the steps above with the LOAD command instead of SAVE.

But now some of you supersleuths are saying, "Aha! All I have to do is type LOAD "*",8 to automatically load the file (assuming it is the first file on the disk)." Mr. Powell is one step ahead of you. He suggests saving a dummy file first on the disk. Type NEW and then SAVE ",",8. Then save your supersensitive program as described above. You would be wise to experiment with these ideas on a newly formatted disk. (We can't take responsibility if you protect your files so well that even you can't access them!) Thanks for the tips, Mr. Powell.

There were several good but lengthy solutions to *Problem #12-1: Phone Fun.* Rather than print the programs, we'll show some of the results. James Speers (Niles, MI) found an appropriate number for me to be 378-7378 (DRUPERT). Bill Sears (Owings Mills, MD) suggested calling 788-2929 (PUT-AWAY) if you are worried about spending so much time writing programs like these.

John Gary (Lafayette, LA) sent some prominent people and phone numbers to match their occupations or habits from a puzzle in his local newspaper. See if you can figure these out: Johnny Carson–866-4448, Ed McMahon–424-2422, and Reggie Jackson–466-3786. Dick O'Donnell (Plattsmouth, NE) also sent a solution to this problem.

Problem #12-2: More Phone Fun was quite a bit easier to program than the first problem. Jim Speers was the only reader who sent a working solution to this problem. His program is listed here.

```
3 REM SOLUTION TO PROBLEM #12-2:
4 REM        MORE PHONE FUN
5 REM SUBMITTED BY
6 REM   JIM SPEERS
10 A$="ABCDEFGHIJKLMNOPRSTUVWXY"
20 B$="222333444555666777888999"
30 INPUT"WORD";W$:L=LEN(W$):PRINT TAB(6)
40 FORI=1 TO L:X$=MID$(W$,I,1):IFX$="Q"
OR X$="Z" THEN 100
50 FOR J=1 TO 24:IF X$=MID$(A$,J,1) THEN
 PRINT MID$(B$,J,1);:GOTO 80
60 NEXT J
70 PRINT X$;
80 NEXT I : END
100 PRINT:PRINT"SORRY, THE LINE'S BUSY"
```

Douglas Sargent (Santiago, CHILE), Rick Koppenhaver (Paramus, NJ), Allen Brewer (Sidney, MT) and Fred Ransom (Oxnard, CA) all sent similar solutions to *Problem #12-3: Letter Slalom.* Their one-liners, with some variations, looked like this:

```
1 FORX=1TO38:PRINTTAB(X)"A":NEXT:FORX=39
TO0STEP-1:PRINTSPC(X)"A":NEXT:GOTO1
```

Hugh Rountree (Perry, FL) sent a solution that displayed sequential letters of the alphabet across the screen as they slalomed down.

The biggest response this month was for solutions to *Problem #12-4: Quick Hex.* Certainly the most unique solution was sent by Dana and Cecil Rousseau (Memphis, TN). They used the LOGO language to define the HEX procedure. Here's their program:

```
TO HEX :N
     MAKE "NUMERAL "
     EXPAND :N
END

TO EXPAND :N
     MAKE "D DIGIT REMAINDER :N 16
     MAKE "NUMERAL WORD :D :NUMERAL
```

```
     IF :N <16 OP :NUMERAL
     EXPAND INTEGER :N / 16
END

TO DIGIT :R
     IF :R < 10 OP :R
     OP CHAR ( :R + 55 )
END
```

To use their program to find the hexadecimal equivalent of 1000, they type HEX 1000 <return>. The C-64 responds with RESULT: 3E8. Notice that the HEX procedure uses the EXPAND procedure which in turn uses the DIGIT procedure. Notice that EXPAND is used recursively, each time with a value which is a factor of 16 less than the previous time. A most interesting solution!

The other submittals were certainly more mundane, but shorter and somewhat more obscure. There were one-liners, two-liners, and many-liners. Of the one-liners, the solution from Werner Hager (Simi Valley, CA) was the most versatile. It allows decimal values up to 4,294,967,296 (16 to the 8th). It can be typed without abbreviations.

```
1 INPUTN:FORI=8TO0STEP-1:Z=INT(N/16^I):N
=N-Z*16^I:PRINTCHR$(Z+48-7*(Z>9));:NEXT
```

The other programs used variations of two or three different algorithms. R.D. Croucher Jr. (Vancouver, WA), Stephen Meirowsky (Wichita, KS), and Jim Speers (Niles, MI) sent one-liners.

Longer programs which used similar ideas but distributed over several lines came from David Breeding (Fairfax, VA), Tony Solomon (Panama City, FL), John Prager (Bay City, MI), Michael Johnston (Haskell, TX), Maurice Tift (Auburn, AL), Hugh Rountree (Perry, FL), Mike Cancilla (Bowmanville, ONT), Dave Antonison (Cambridge, ONT), and Charles A. Wilson (Whitesboro, NY).

The final listing is the solution from Fred Ransom (Oxnard, CA). Mr. Ransom's program displays fractional results as well as integer values. For example, the value 1000.5 in hexadecimal is given as 03E8.8, and 198.02 is 00C6.051EB8.

```
3 REM SOLUTION TO PROBLEM #12-4:
4 REM      QUICK    HEX
5 REM SUBMITTED BY FRED RANSOM
6 REM HANDLES FRACTIONS AND INTEGERS
10 INPUT"NUMBER";A:IFA<0 OR A>65535 THEN
 END
20 FORI=3TO0STEP-1:B=INT(A/16^I):B=B-INT
(B/16)*16+48:IFB>=58THENB=B+7
30 PRINTCHR$(B);:NEXT:PRINT".";
40 A=A-INT(A):IFA=0THENPRINT:GOTO10
50 A=A*16:B=INT(A)+48:IFB>57THENB=B+7
60 PRINTCHR$(B);:GOTO40
```

Congratulations to the others who sent solutions to previous months' *Commodares* including Michael Heindel (York, PA), David Rice (Phoenix, AZ), Derward McKinney (APO NY), Dan Bachmann (Bridgewater, NJ), Steven Evans (Los Alamitos, CA), and Robert D. Abernethy (APO NY). Be ready for more problems next month! □

COMMODORE ROOTS

Continued from page 56

RUNNING THE *HI.TEST* PROGRAM

Now that you know what supplies you'll need for next month's column, I'm ready to show you how you can write and run a machine language program—without having to use an assembler. One way to do that is to invoke a series of POKE statements from within a BASIC program.

To run a machine language program using this technique, all you have to do is convert the machine language program into a series of decimal numbers, and then POKE each of those numbers into a series of free memory registers. Then you can use the SYS command in Commodore BASIC to execute your machine language program.

We'll do that with the *HI.TEST* program right now. Here's the *HI.TEST* program, converted into decimal numbers, translated into BASIC, and ready to be run as a BASIC program:

HI.TEST.BAS
(HI.TEST Program, Basic Version)

```
10 REM *** HI.TEST.BAS ***
20 DATA 169,72,32,210,255,169,73,32,210,
255,96
30 FOR L = 49152 TO 49162:READ X:POKE L,
X:NEXT L
40 SYS 49152
```

Here's how the *HI.TEST.BAS* program works:

Line 20 is a line of data that equates to a series of machine language instructions. In Line 30, there is a loop that will POKE each numeric instruction in Line 20 into a block of RAM that extends from Memory Address 49152 to Memory Address 49162 ($C000 to $C010 in hexadecimal notation). Finally, in Line 40, there's a SYS command that executes the machine language program.

You can type and run *HI.TEST.BAS* just as you would any BASIC program. And almost any machine language program can be converted into BASIC and run in this manner. □

Next month, *By the Numbers*—an easy introduction to bits, bytes, and binary numbers. If penguins could count, why they'd count in binary numbers. Why you'd count in hexadecimal numbers if you had sixteen fingers. A special bonus: a type and run program that will convert numbers from one base to another, totally painlessly.

Roll Over Pachelbel

Classical Music for the Commodore 64

By Simon Edgeworth

The composer Pachelbel would be surprised to hear this version of his famous "Canon." And he certainly would not believe that the orchestra was a tiny piece of silicon called the SID chip.

This program plays 12 different verses of "Pachelbel's Canon," using all three of your Commodore 64's music voices. It uses an unusual technique to do this. The melody data contains only seven simple tunes. To play a verse, the program chooses up to three of these simple tunes and plays them together. A different combination of up to three tunes is used for each verse. This enables the program to play for a long time without the hundreds of data lines that would otherwise be needed.

HOW IT WORKS

The melody data (lines 400-474) is written in an easy-to-understand format. For example, the note "C" in the third octave is simply written "C3". A zero is used when no note is to be played. The routine at lines 20-44 translates the melodies into frequencies, and stores them in a safe area of memory.

The actual music-making is done by lines 100-126. At the beginning of each verse, one line is read from the plan (lines 700-732), and this determines which tunes will be played. For example, when the program plays the fourth verse, it reads line 708, and plays melodies one, two, and four simultaneously. A zero means that one of the instruments is silent, and −1 signals the end of the program.

MAKING CHANGES

If you're feeling adventurous, you can change the music in various ways.

1. Try changing the first number (16) on line 322. A higher number will slow the music down, and a lower number will speed it up.

2. Try changing or adding to the plan (lines 700-732).

3. You can alter the sound of the three instruments by changing lines 602-606. The first character is the waveform, and can be T, S, P, or N (triangle, sawtooth, pulse, or noise). The next four numbers are attack, decay, sustain, and release (0-15). The final number is the pulse frequency (0-4096). □

SEE PROGRAM LISTING ON PAGE 113

If you'd like to spare yourself the time and trouble of typing in *Roll Over Pachelbel* and all the other outstanding programs in this issue of *Ahoy!*, they're all available on our monthly disk or cassette. See page 33 for details.

BASIC Trace

Continued from page 57

HOW IT WORKS

The key to my implementation of *Trace* is in the use of a Wedge, and in memory locations 57 and 58. If you look on a memory map of the Commodore 64, you will see that by locations 57-58 it says "Current BASIC Line Number" or something to that effect. What this means is that anytime you PEEK locations 57 and 58, you will know what line of your program the computer is currently executing. If the computer is not running your BASIC program at the time you PEEK, then you will get either the last line number that was run or garbage from these locations. Why, you might ask, do you have to PEEK *two* locations to find the current line number? The reason is that each location can only hold a number from 0 to 255. We all have written or seen programs with line numbers much greater than 255, though. Therefore, to accommodate these larger numbers, the computer chains two memory locations together in a low byte/high byte format. To convert this to normal decimal numbers, take the contents of the second location, the high byte (in our case, loc. 58), and multiply it by 256. Then add to it the contents of the first location, the low byte (in our case, loc. 57). The resulting number will be the line of the BASIC program that is currently being executed. A formula for this is: Line Number = PEEK(58) * 256 + PEEK(57). Thus by using locations 57 and 58, one can readily find out what line of a BASIC program is executing.

Now that we know how to find out what line we are on, we must be able to continually place that information on the screen while *at the same time* a BASIC program is running. This is not that difficult if you use a wedge. A wedge literally means that "you wedge your program into somebody else's program." If you look at a memory map again, you will see that in locations 115- 138 there is something called the CHRGET subroutine. This routine is called by the BASIC ROM every time that you hit the RETURN key, or every time a BASIC program line is executed. Thus what I did was wedge my program into the CHRGET subroutine. Of course, I couldn't really put my program inside theirs, because there is not enough memory for it, but I had CHRGET jump to my routine, and then when I am done, I just jump back to CHRGET.

The result of this is that my program runs whenever the RETURN key is hit or whenever a BASIC program is running. While my program runs, it reads locations 57 and 58 to figure out which line you are on, puts this line into the trace window, and reprints the window onto the screen. In this way, the program can tell you every BASIC line that is executed.

CONCLUSION

The *Trace* program is very useful as a debugging tool for determining the flow of a BASIC program. By watching the compact window display one can see the major branches that the program is taking, which will be very valuable in fixing a faulty BASIC program. By using the English command, TRACE, to invoke the program, it should be hard to forget how to use this utility. ☐

SEE PROGRAM LISTING ON PAGE 107

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listing guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!*'s program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart. The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: 22 spaces, for example, as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines, long; VIC 20 lines, a maximum of 88 characters, or 4 screen lines). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the VIC 20 and C-64. The version appropriate for your machine will help you proofread our programs after you type them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See the instructions preceding each program.) □

When You See	It Means	You Type	You Will See
[CLEAR]	Screen Clear	SHIFT CLR/HOME	
[HOME]	Home	CLR/HOME	
[UP]	Cursor Up	SHIFT ↑CRSR↓	
[DOWN]	Cursor Down	↑CRSR↓	
[LEFT]	Cursor Left	SHIFT ←CRSR→	
[RIGHT]	Cursor Right	←CRSR→	
[SS]	Shifted Space	SHIFT Space	
[INSERT]	Insert	SHIFT INST/DEL	
[DEL]	Delete	INST/DEL	
[RVSON]	Reverse On	CNTRL 9	
[RVSOFF]	Reverse Off	CNTRL 0	
[UPARROW]	Up Arrow	↑	
[BACKARROW]	Back Arrow	←	
[PI]	PI	π	
[EP]	English Pound	£	

When You See	It Means	You Type	You Will See
[BLACK]	Black	CNTRL 1	
[WHITE]	White	CNTRL 2	
[RED]	Red	CNTRL 3	
[CYAN]	Cyan	CNTRL 4	
[PURPLE]	Purple	CNTRL 5	
[GREEN]	Green	CNTRL 6	
[BLUE]	Blue	CNTRL 7	
[YELLOW]	Yellow	CNTRL 8	
[F1]	Function 1	F1	
[F2]	Function 2	SHIFT F1	
[F3]	Function 3	F3	
[F4]	Function 4	SHIFT F3	
[F5]	Function 5	F5	
[F6]	Function 6	SHIFT F5	
[F7]	Function 7	F7	
[F8]	Function 8	SHIFT F7	

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** This and the preceding page explain these codes and provide other essential information on entering **Ahoy!** programs. Read these pages **before** entering programs.

BUG REPELLENT

This program will let you debug any *Ahoy!* program. Follow instructions for VIC 20 (cassette or disk) or C-64.

VIC 20 VERSION

By Michael Kleinert and David Barron

For cassette: type in and save the *Bug Repellent* program, then type RUN 63000[RETURN]SYS 828[RETURN]. If you typed the program properly, it will generate a set of two-letter line codes that will match those listed to the right of the respective program lines.

Once you've got a working *Bug Repellent*, type in the program you wish to check. Save it and type the RUN and SYS commands listed above once again, then compare the line codes generated to those listed in the magazine. If you spot a discrepancy, a typing error exists in that line. Important: you must use exactly the same spacing as the program in the magazine. Due to memory limitations on the VIC, the VIC *Bug Repellent* will register an error if your spacing varies from what's printed.

You may type SYS 828 as many times as you wish, but if you use the cassette for anything, type RUN 63000 to restore the *Repellent*.

When your program has been disinfected you may delete all lines from 63000 on. (Be sure the program you type doesn't include lines above 63000!)

For disk: enter *Bug Repellent*, save it, and type RUN:NEW [RETURN]. Type in the program you wish to check, then SYS 828.

To pause the line codes listing, press SHIFT.

To send the list to the printer type OPEN 4,4:CMD 4:SYS 828[RETURN]. When the cursor comes back, type PRINT#4:CLOSE 4[RETURN].

```
·63000 FORX=828TO1023:READY:POKEX,Y:NEXT:END   AC
·63001 DATA169,0,133,63,133,64,165,43,133,251  JL
·63002 DATA165,44,133,252,160,0,132,254,32,228 DF
·63003 DATA3,234,177,251,208,3,76,208,3,230    OE
·63004 DATA251,208,2,230,252,169,244,160,3,32  OH
·63005 DATA30,203,160,0,177,251,170,230,251,20
 8                                             KO
·63006 DATA2,230,252,177,251,32,205,221,169,58 JJ
·63007 DATA32,210,255,169,0,133,253,230,254,32 OK
·63008 DATA228,3,234,165,253,160,0,170,177,251 LG
·63009 DATA201,32,240,6,138,113,251,69,254,170 BP
·63010 DATA138,133,253,177,251,208,226,165,253
 ,41                                           DD
·63011 DATA240,74,74,74,74,24,105,65,32,210    EK
·63012 DATA255,165,253,41,15,24,105,65,32,210  FO
·63013 DATA255,169,13,32,210,255,173,141,2,41  PK
·63014 DATA1,208,249,230,63,208,2,230,64,230   CB
·63015 DATA251,208,2,230,252,76,74,3,169,236   KH
·63016 DATA160,3,32,30,203,166,63,165,64,32    DP
·63017 DATA205,221,169,13,32,210,255,96,230,25
 1                                             EL
·63018 DATA208,2,230,252,96,0,76,73,78,69      OI
·63019 DATA83,58,32,0,76,73,78,69,32,35        FG
·63020 DATA32,0,0,0,0,0                        LE
```

C-64 VERSION

By Michael Kleinert and David Barron

Type in, SAVE, and RUN the *Bug Repellent*. Type NEW, then type in or LOAD the *Ahoy!* program you wish to check. When that's done, SAVE your program (don't RUN it!) and type SYS 49152 [RETURN].

To pause the listing depress and hold the SHIFT key.

Compare the codes your machine generates to the codes listed to the right of the respective program lines. If you spot a difference, an error exists in that line. Jot down the number of lines where contradictions occur, LIST each line, spot the errors, and correct them.

```
·5000 FORX=49152TO49488:READY:POKEX,Y:NEXT:END GJ
·5001 DATA32,161,192,165,43,133,251,165,44,133 DL
·5002 DATA252,160,0,132,254,32,37,193,234,177  DB
·5003 DATA251,208,3,76,138,192,230,251,208,2   OF
·5004 DATA230,252,76,43,192,76,73,78,69,32     KN
·5005 DATA35,32,0,169,35,160,192,32,30,171     CA
·5006 DATA160,0,177,251,170,230,251,208,2,230  CE
·5007 DATA252,177,251,32,205,189,169,58,32,210 JE
·5008 DATA255,169,0,133,253,230,254,32,37,193  CL
·5009 DATA234,165,253,160,0,76,13,193,133,253  NB
·5010 DATA177,251,208,237,165,253,41,240,74,74 MB
·5011 DATA74,74,24,105,65,32,210,255,165,253   EP
·5012 DATA41,15,24,105,65,32,210,255,169,13    GH
·5013 DATA32,220,192,230,63,208,2,230,64,230   AN
·5014 DATA251,208,2,230,252,76,11,192,169,153  NG
·5015 DATA160,192,32,30,171,166,63,165,64,76   BF
·5016 DATA231,192,96,76,73,78,69,83,58,32      EP
·5017 DATA0,169,247,160,192,32,30,171,169,3    PJ
·5018 DATA133,254,32,228,255,201,83,240,6,201  FK
·5019 DATA80,208,245,230,254,32,210,255,169,4  FL
·5020 DATA166,254,160,255,32,186,255,169,0,133 CL
·5021 DATA63,133,64,133,2,32,189,255,32,192    GC
·5022 DATA255,166,254,32,201,255,76,73,193,96  NN
·5023 DATA32,210,255,173,141,2,41,1,208,249    NH
·5024 DATA96,32,205,189,169,13,32,210,255,32   IM
·5025 DATA204,255,169,4,76,195,255,147,83,67   KC
·5026 DATA82,69,69,78,32,79,82,32,80,82        DC
·5027 DATA73,78,84,69,82,32,63,32,0,76         ML
·5028 DATA44,193,234,177,251,201,32,240,6,138  GN
·5029 DATA113,251,69,254,170,138,76,88,192,0   JK
·5030 DATA0,0,0,230,251,208,2,230,252,96       NA
·5031 DATA170,177,251,201,34,208,6,165,2,73    DM
·5032 DATA255,133,2,165,2,208,218,177,251,201  JA
·5033 DATA32,208,212,198,254,76,29,193,0,169   FM
·5034 DATA13,76,210,255,0,0,0                  PA
```

FLANKSPEED FOR THE C-64

By Gordon F. Wheat

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1 – SAVEs what you have entered so far.

f3 – LOADs in a program worked on previously.

f5 – To continue on a line you stopped on after LOADing in the previously saved work.

f7 – Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program. f7 temporarily freezes the output as well.

```
·5 POKE53280,12:POKE53281,11 ·                 LL
·6 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[
 15" "]";                                      ED
·10 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY P
 ROGRAM[6" "]"                                 MC
·15 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[
 9" "]"                                        DM
·20 PRINT"[RVSON][3" "]COPR. 1984, ION INTERNA
```

```
TIONAL INC.[3" "]"                                   DH
30 FORA=54272TO54296:POKEA,0:NEXT                    IM
40 POKE54272,4:POKE54273,48:POKE54277,0:POKE5
4278,249:POKE54296,15                                NH
70 FORA=680TO699:READB:POKEA,B:NEXT                  KO
75 DATA169,251,166,253,164,254,32,216,255,96         HJ
76 DATA169,0,166,251,164,252,32,213,255,96           JB
80 B$="STARTING ADDRESS IN HEX":GOSUB2010:AD=
B:SR=B                                               HC
85 GOSUB2520:IFB=0THEN80                             FO
86 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16         KE
90 B$="ENDING ADDRESS IN HEX":GOSUB2010:EN=B         IF
95 GOSUB2510:IFB=0THEN80                             FP
96 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16             MN
97 IFB>255THENB=B-255:POKE254,PEEK(254)+1            GE
98 POKE253,B:PRINT                                   HN
100 REM  GET HEX LINE                                IL
110 GOSUB3010:PRINT": [c P][LEFT]";:FORA=0TO8        FG
120 FORB=0TO1:GOTO210                                MD
125 NEXTB                                            ME
130 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN310            LH
135 PRINT" [c P][LEFT]";                             IK
140 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "            PD
150 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255           LK
160 NEXT                                             IA
170 IFA%(8)<>TTHENGOSUB1010:GOTO110                  FK
180 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOT
0110                                                 MN
200 REM    GET HEX INPUT                             AB
210 GETA$:IFA$=""THEN210                             HO
211 IFA$=CHR$(20)THEN270                             GC
212 IFA$=CHR$(133)THEN4000                           MD
213 IFA$=CHR$(134)THEN4100                           KF
214 IFA$=CHR$(135)THENPRINT" ":GOTO4500              GE
215 IFA$=CHR$(136)THENPRINT" ":GOTO4700              BJ
220 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO
250                                                  GM
230 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO
250                                                  LE
240 GOSUB1100:GOTO210                                LL
250 PRINTA$"[c P][LEFT]";                            OA
260 GOTO125                                          CG
270 IFA>0THEN280                                     OP
272 A=-1:IFB=1THEN290                                OB
274 GOTO140                                          CJ
280 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1           HG
285 A=A-1                                            BE
290 PRINTCHR$(20);:GOTO140                           KH
300 REM LAST LINE                                    AD
310 PRINT" ":T=AD-(INT(AD/256)*256)                  GJ
320 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255         PL
330 NEXT                                             IA
340 IFA%(A)<>TTHENGOSUB1010:GOTO110                  KF
350 FORB=0TOA-1:POKEAD+B,A%(B):NEXT                  HN
360 PRINT:PRINT"YOU ARE FINISHED!":GOTO4000          ON
1000 REM BELL AND ERROR MESSAGES                     FL
1010 PRINT:PRINT"LINE ENTERED INCORRECTLY":PR
INT:GOTO1100                                         DH
1020 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":
GOTO1100                                             JA
1030 PRINT:PRINT"ENDING IS LESS THAN STARTING
!":B=0:GOTO1100                                      HD
1040 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED
RANGE!":B=0:GOTO1100                                 AG
1050 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:G
OTO1100                                              KN
1060 PRINT"?ERROR IN SAVE":GOTO1100                  EI
1070 PRINT"?ERROR IN LOAD":GOTO1100                  GL
1080 PRINT:PRINT:PRINT"END OF ML AREA":PRINT         PG
1100 POKE54276,17:POKE54276,16:RETURN                BH
1200 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA
$:RETURN                                             IM
2000 REM GET FOUR DIGIT HEX                          PC
2010 PRINT:PRINTB$;:INPUTT$                          GM
2020 IFLEN(T$)<>4THENGOSUB1020:GOTO2010              II
2040 FORA=1TO4:A$=MID$(T$,A,1):GOSUB2060:IFT(
A)=16THENGOSUB1020:GOTO2010                          AD
2050 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+
T(4):RETURN                                          GF
2060 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RET
URN                                                  EH
2070 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RET
URN                                                  KP
2080 T(A)=16:RETURN                                  NP
2500 REM ADRESS CHECK                                LI
2510 IFAD>ENTHEN1030                                 MI
2515 IFB<SRORB>ENTHEN1040                            MG
2520 IFB<256OR(B>40960ANDB<49152)ORB>53247THE
N1050                                                MI
2530 RETURN                                          IM
3000 REM ADDRESS TO HEX                              EB
3010 AC=AD:A=4096:GOSUB3070                          HG
3020 A=256:GOSUB3070                                 CE
3030 A=16:GOSUB3070                                  PN
3040 A=1:GOSUB3070                                   MJ
3060 RETURN                                          IM
3070 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO3
090                                                  CJ
3080 A$=CHR$(T+48)                                   JP
3090 PRINTA$;:AC=AC-A*T:RETURN                       AC
4000 A$="**SAVE**":GOSUB4200                         AI
4050 OPEN1,T,1,A$:SYS680:CLOSE1                      LH
4060 IFST=0THENEND                                   EO
4070 GOSUB1060:IFT=8THENGOSUB1200                    FJ
4080 GOTO4000                                        FF
4100 A$="**LOAD**":GOSUB4200                         AB
4150 OPEN1,T,0,A$:SYS690:CLOSE1                      MF
4160 IFST=64THEN110                                  JH
4170 GOSUB1070:IFT=8THENGOSUB1200                    CM
4180 GOTO4100                                        FO
4200 PRINT" ":PRINTTAB(14)A$                         FG
4210 PRINT:A$="":INPUT"FILENAME";A$                  OM
4215 IFA$=""THEN4210                                 GF
4220 PRINT:PRINT"TAPE OR DISK?":PRINT                DF
4230 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RE
TURN                                                 IG
4240 IFB$<>"T"THEN4230                               FN
4250 RETURN                                          IM
4500 B$="CONTINUE FROM ADDRESS":GOSUB2010:AD=
B                                                    DK
4510 GOSUB2515:IFB=0THEN4500                         MA
4520 PRINT:GOTO110                                   OI
4700 B$="BEGIN SCAN AT ADDRESS":GOSUB2010:AD=
B                                                    FH
4705 GOSUB2515:IFB=0THEN4700                         NK
4706 PRINT:GOTO4740                                  DI
4710 FORB=0TO7:AC=PEEK(AD+B):GOSUB3030:IFAD+B
=ENTHENAD=SR:GOSUB1080:GOTO110                       BK
4715 PRINT" ";:NEXTB                                 EC
4720 PRINT:AD=AD+8                                   GN
4730 GETB$:IFB$=CHR$(136)THEN110                     MN
4740 GOSUB3010:PRINT": ";:GOTO4710                   JD
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 93 and 94 explain these codes and provide other essential information on entering *Ahoy!* programs. Refer to these pages **before** entering any programs!

The Joy of Sticks

FROM PAGE 18

C-64 VERSION

```
•1 REM COMMODORE 64 JOYSTICK                      CP
•2 REM                                            JD
•3 REM USE SECOND JOYSTICK PORT                   EJ
•4 REM                                            JD
•10 GOSUB 600:GOSUB 500                           IP
•80 V=10:H=10                                     PG
•90 GOTO 140                                      CJ
•97 REM                                           JD
•98 REM MAIN MOVEMENT LOOP                        EH
•99 REM                                           JD
•100 S=PEEK(56320)                                CM
•101 IF (S AND 1)=0 THEN V=V-1:IF V<0 THE
 N V=V+BF                                         CD
•102 IF (S AND 2)=0 THEN V=V+1:IF V>BE TH
 EN V=V-BF                                        JM
•103 IF (S AND 4)=0 THEN H=H-1:IF H<0 THE
 N H=H+RF                                         LA
•104 IF (S AND 8)=0 THEN H=H+1:IF H>RE TH
 EN H=H-RF                                        EM
•105 IF (S AND 16)=0 THEN END                     FP
•110 IF H=HX AND V=VX THEN 100                    OK
•140 PRINT VM$(VX)TAB(HX)" "VM$(V)TAB(H)P
 F$                                               JD
•150 HX=H:VX=V                                    IM
•190 GOTO 100                                     CF
•497 REM                                          JD
•498 REM SET UP MOVEMENT STRINGS                  LP
•499 REM                                          JD
•500 DIM VM$(BE):VM$(0)=CHR$(19)                  PF
•510 FOR I=1 TO BE:VM$(I)=VM$(I-1)+CHR$(1
 7):NEXT I                                        CG
•540 RETURN                                       IM
•597 REM                                          JD
•598 REM SET UP SCREEN MARGINS                    EO
•599 REM                                          JD
•600 RE=38                                        LM
•605 RF=39                                        LM
•610 BE=23                                        KK
•615 BF=24                                        KK
•617 REM                                          JD
•618 REM PLAYER FIGURE                            EO
•619 REM                                          JD
•620 PF$=CHR$(122)                                BN
•627 REM                                          JD
•628 REM SCREEN COLOR                             MP
•629 REM                                          JD
•630 POKE 53281,0:POKE 53280,0:PRINT CHR$
 (147)                                            II
•690 RETURN                                       IM
```

VIC 20 VERSION

```
•1 REM VIC JOYSTICK                               MP
•2 REM                                            JD
•3 REM STOP THIS PROGRAM BY PRESSING JOYS
 TICK BUTTON!!!                                   PM
•4 REM                                            JD
•10 GOSUB 600:GOSUB 500                           IO
•80 V=10:H=10                                     PG
•90 GOTO 140                                      CL
•97 REM                                           JD
•98 REM MAIN MOVEMENT LOOP                        DI
•99 REM                                           JD
•100 S=(PEEK(37137)AND 60)OR (PEEK(37152)
 AND 128)                                         KE
•101 IF (S AND 4)=0 THEN V=V-1                    NF
•102 IF (S AND 8)=0 THEN V=V+1                    LO
•103 IF (S AND 16)=0 THEN H=H-1                   LC
•104 IF (S AND 128)=0 THEN H=H+1                  CK
•105 IF (S AND 32)=0 THEN POKE 37154,255:
 END                                              KJ
•110 IF H=HX AND V=VX THEN 100                    OC
•120 IF V>BE THEN V=V-BF                          IP
•121 IF V<0 THEN V=V+BF                           PO
•122 IF H>RE THEN H=H-RF                          EF
•123 IF H<0 THEN H=H+RF                           PA
•140 PRINT VM$(VX)TAB(HX)" "VM$(V)TAB(H)P
 F$                                               JP
•150 HX=H:VX=V                                    IM
•190 GOTO 100                                     BP
•497 REM                                          JD
•498 REM SET UP MOVEMENT STRINGS                  BA
•499 REM                                          JD
•500 DIM VM$(BE):VM$(0)=CHR$(19)                  CO
•510 FOR I=1 TO BE:VM$(I)=VM$(I-1)+CHR$(1
 7):NEXT I                                        DN
•540 RETURN                                       IM
•597 REM                                          JD
•598 REM SET UP SCREEN MARGINS                    PP
•599 REM                                          JD
•600 RE=20                                        LF
•605 RF=21                                        LF
•610 BE=21                                        KE
•615 BF=22                                        KE
•620 PF$=CHR$(122):REM PLAYER-FIGURE              NO
•627 REM                                          JD
•628 REM SCREEN COLOR                             PD
•629 REM                                          JD
•630 POKE 36879,8:PRINT CHR$(147)                 GN
•660 POKE 37154,127:REM ENABLE JOYSTICK (
 DISABLES PART OF KEYBOARD)                       CD
•690 RETURN                                       IM
```

Getting into the Kernal

FROM PAGE 45

```
•10 REM - RUPERT REPORT LISTING 1 -               IN
•20 REM                                           JD
•30 REM -- SAVE MEMORY TO DISK --                 CF
•39 REM                                           JD
```

```
•40 REM * * * * * * * * * * * * * * * * *    GD
•41 REM *                               *    OO
•42 REM *  NOTE THAT LINES 180 - 400    *    IK
•43 REM *   MAY BE OMITTED IF THESE     *    LN
•44 REM *   TWO LINES ARE CHANGED:      *    HE
•45 REM *     200 OPEN 8,8,8,FLNAM$     *    JN
•46 REM *     520 CLOSE 8               *    DH
•47 REM *                               *    OO
•48 REM * * * * * * * * * * * * * * * * *    GD
•49 REM                                      JD
•50 PRINT CHR$(147) :REM CLEAR-SCRN          AN
•60 DEF FNH(X)=INT(X/256)                    IB
•70 DEF FNL(X)=X-256*FNH(X)                  LC
•80 AREG=780 : XREG=781 : YREG=782           HE
•90 REM                                      JD
•100 REM<><><><><><><><><><><><><><><><>     HD
•110 REM PUT FILENAME, STARTING ADDRESS      AE
•120 REM     AND ENDING ADDRESS BELOW        PC
•130 REM<><><><><><><><><><><><><><><><><>   HD
•140 FLNAM$="DOS 5.1" :REM   FILENAME        BL
•150 SADDR=52224 :REM $CC00 (START ADDR)     CC
•160 EADDR=53082 :REM $CF5A (END ADDR+1)     MH
•170 REM                                     JD
•180 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•190 REM ...KERNAL ROUTINE: SETLFS...        AC
•200 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•210 SLFS=65466  :REM $FFBA                  PH
•220 FLNUM=8 : DNUM=8 : CMMD=255             IE
•230 POKE AREG,FLNUM                         HB
•240 POKE XREG,DNUM : POKE YREG,CMMD         HI
•250 SYS SLFS                                NG
•260 PRINT"BACK FROM 'SETLFS'":PRINT         HF
•270 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•280 REM ...KERNAL ROUTINE: SETNAM...        JH
•290 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•300 SNAM=65469  :REM $FFBD                  PO
•310 BUFF=49155  :REM $C003                  JK
•320 LNAM=LEN(FLNAM$)                        HM
•330 FOR N=1 TO LNAM                         IA
•340 POKE BUFF+N-1,ASC(MID$(FLNAM$,N,1))     LF
•350 NEXT N                                  NA
•360 POKE AREG,LNAM                          BK
•370 POKE XREG,FNL(BUFF)                     HI
•380 POKE YREG,FNH(BUFF)                     KD
•390 SYS SNAM                                ML
•400 PRINT"BACK FROM 'SETNAM'":PRINT         FK
•410 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•420 REM ...KERNAL ROUTINE: SAVE...          BD
•430 REM=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-   DD
•440 SVE=65496   :REM $FFD8                  ME
•450 ZPTR=251    :REM $FB (ZERO PAGE)        JG
•460 POKE ZPTR,FNL(SADDR)                    NO
•470 POKE ZPTR+1,FNH(SADDR)                  MK
•480 POKE AREG,ZPTR                          DE
•490 POKE XREG,FNL(EADDR)                    KK
•500 POKE YREG,FNH(EADDR)                    KN
•510 SYS SVE                                 IL
•520 PRINT"BACK FROM 'SAVE'":PRINT           MM
•530 PRINT"[3"="]MEMORY SAVE ROUTINE[3"="
 ]"                                          BM
•540 PRINT"FILE NAME:[8" "]";FLNAM$          FH
•550 PRINT"STARTING ADDRESS: ";SADDR         NH
•560 PRINT"ENDING ADDRESS:[3" "]";EADDR-1    MP
•570 PRINT                                   JJ
```

HOP AROUND

FROM PAGE 86 **C-64 VERSION**

```
•1 Y=1:PRINT"[CLEAR]"                        OH
•2 GOTO80                                    PA
•3 E=+1:POKEA,32:POKEA+H,32:A=A+E:IFPEEK(
 A+80)=32THEN7                               ID
•4 GOTO37                                    PC
•5 E=-1:POKEA,32:POKEA+H,32:A=A+E:IFPEEK(
 A+80)=32THEN7                               GG
•6 GOTO37                                    PC
•7 L=.:RESTORE                               PI
•8 POKEA,32:POKEA+H,32:L=L+1:A=A+H:IFPEEK
 (A+80)=4THEN12                              AH
•9 IFA=BORA>1944THEN49                       HI
•10 READTW,T:POKEA,.:POKEA+H,1:POKES+1,TW
 :POKES,T:GOTO8                              FM
•12 IFL>=9THEN55                             HP
•13 POKES+1,0:POKES,0:GOTO37                 OL
•14 RESTORE:FORTT=1TO21:READTW,T:NEXT:BJ=
 BJ+1                                        MK
•15 READTW,T:POKEA,32:POKEA+40,32:IFPEEK(
 A-40)=4THENA=A+1:GOTO7                      KD
•17 POKES+1,TW:POKES,T:IFA<1064THENA=A+1:
 GOTO7                                       JM
•18 IFPEEK(A-40)<7THEN55                     NJ
•19 A=A-40:POKEA,.:POKEA+40,1:GOTO15         LO
•20 POKE53281,1:POKE53280,2:PRINT"[CLEAR]
 [4"[DOWN]"][15"[RIGHT]"][RED]HOP AROUND!
 "                                           LO
•21 PRINT"[15"[RIGHT]"][11"[c T]"]":PRINT
 "[DOWN][RIGHT][BLUE][12"[RIGHT]"]BY KEVI
 N DEWEY"                                    LH
•22 PRINT"[DOWN][GREEN][14"[RIGHT]"]USE J
 OYSTICK[DOWN][12"[LEFT]"]FOR MOVEMENT"      HG
•23 PRINT"[DOWN][DOWN][BLACK][14"[RIGHT]"
 ]HIT ANY KEY"                               HA
•24 GOTO600                                  CC
•26 POKEVV+21,0:GOTO58                       OH
•27 GU=1:POKEB-40,2:GOTO55                   CN
•30 A=1797:B=1065:S=54272:POKES+5,200:POK
 ES+6,200:POKES+24,15:LL=56320:C=1142        DL
•31 MZ=7:FORT=STOS+24:POKET,0:NEXT:POKES+
 24,15:POKES+3,8:POKES+2,0:POKES+5,144       KN
•32 H=40:PRINT"[HOME][13"[DOWN]"][40"D"]"
 ;                                           PB
•34 POKEB,2:POKEB+40,3:D=PEEK(LL):IFD=123
 THEN5                                       JF
•35 POKEC-40,6:POKEC,MZ:IFD=119THEN3         JM
•36 IFD=111THEN14                            CG
```

```
•37 POKEA,.:POKEA+40,1:IFPEEK(A+80)=32THE
N7                                          BP
•38 M=INT(RND(1)*9)+1:IFM=9ANDP=.THEN42     CL
•39 IFA=BTHEN27                             ND
•40 IFP=1THENRETURN                         PH
•41 GOTO34                                  PF
•42 N=INT(RND(1)*37)+1144:POKES+6,243:POK
ES+1,4:POKES,112                            HE
•43 P=1:FORO=NTON+840STEP40:POKEO,5:GOSUB
34:POKEO,32:O=O+40:IFO=AORO=A+40THEN55      MO
•44 POKES+4,65:IFPEEK(O+40)=4THENPOKES+4,
129:O=O+1                                   FO
•45 POKES+4,33:NEXT:POKES+1,0:POKES,0       LL
•46 P=0:POKEN,32:N=.:POKEB,32:POKEB+40,32
:B=B+Y:IFPEEK(B)=6ORB>1102THEN51            IA
•47 IFMZ=8THENMZ=7:GOTO34                   LF
•48 IFMZ=7THENMZ=8:GOTO34                   JP
•49 IFA=BTHEN64                             NK
•50 GOTO55                                  PO
•51 RESTORE:LA=1062:LP=1904:X=33:POKES+2,
100:POKES+3,100:POKE1102,32:POKE1142,32     CH
•52 FORT=LATOLPSTEP+39:POKET,7:POKET-40,6
:POKET-1,3:POKET-41,2:READW,TT:POKES+4,X    CC
•53 POKES+1,W:POKES,TT:POKES+4,17:POKET,3
2:POKET-40,32:POKET-1,32:POKET-41,32        AI
•54 POKES+4,65:NEXT:GOTO150                 GK
•55 RR=2:POKEA,32:POKEA+40,9:POKEA+41,10:
IFGU=1THENPOKEB,3:GU=.:POKES+5,200          GB
•56 POKES+6,100:POKES+4,129:POKES+1,33:PO
KES,135:FORT=15TO0STEP-1:POKES+24,T         AE
•57 FORTT=1TO30:NEXT:NEXT:POKES+1,0:POKES
,0:FORT=1TO500:NEXT:POKE53272,21:GOTO26     MN
•58 PRINT"[CLEAR][6"[DOWN]"][18"[RIGHT]"]
[BLACK]YOU";:GOSUB63:PRINT"[DOWN][DOWN][
3"[LEFT]"]BOMBED";:GOSUB63                  MC
•59 PRINT"[DOWN][DOWN][3"[LEFT]"]OUT!":GO
SUB63:PRINT"[3"[DOWN]"][13"[RIGHT]"][RED
]HIT ANY KEY"                               BD
•61 IFPEEK(197)<>64THENRUN                  EA
•62 GOTO61                                  PF
•63 FORT=1TO1000:NEXT:RETURN                JD
•64 POKEB-1,.:POKEB+39,1:RESTORE            JI
•65 FORT=BTO1970STEP+42:POKET,2:POKET+40,
3:READTT,TW:POKES,TW:POKES+1,TT             HJ
•66 POKES+4,33:FORTG=1TO15:NEXT:POKET,32:
POKET+40,32:POKES+4,17:NEXT                 EA
•67 FORT=7TO0STEP-1:POKE2040+T,0:POKES+1,
14:POKES,239:POKES+4,129:POKES+24,T*2       CI
•68 POKES+4,33:NEXT:POKES+1,0:POKES,0:POK
EVV+21,0                                    DP
•69 RR=1:POKE53272,21:PRINT"[CLEAR][PURPL
E]YOU DID IT!":PRINT"[3"[DOWN]"][6" "]YO
U SAVED SWEET DORIS"                        NI
•70 PRINT"[6" "]WITH ONLY";BJ;" JUMPS!"     FF
•71 IFBJ>JBTHENPRINT"[DOWN][DOWN][10"[RIG
HT]"]A[DOWN][LEFT]NEW[DOWN][3"[LEFT]"]HI
GH!!":JB=BJ                                 HE
•75 PRINT"[14"[RIGHT]"]HIT ANY KEY."        JH
•76 IFPEEK(197)<>64THENRUN                  EA
•77 GOTO76                                  PP
•78 STOP                                    JC
•80 POKE52,48:POKE56,48:CLR:GOTO100         OJ
•81 FORI=1TO74:READJ:NEXT:FORI=12288TO123
77:READJ:POKEI.J:NEXT                       PL
•82 DATA14,24,13,78,12,143,11,218,11,48,1
0,143,9,247,9,104,8,225,8,97,7,233,7,119    PK
•83 DATA7,12,6,167,6,71,5,237,5,152,5,71,
4,251,4,180,4,112,4,112,4,180,4,251,5,71    LH
•84 DATA5,152,5,237,6,71,6,167,7,12,7,119
,7,233,8,97,8,225,9,104,9,247,10,143        CM
•85 DATA60,126,219,255,231,126,66,60,24,2
55,189,189,60,36,36,102,126,90,126,255      LG
•86 DATA129,195,102,60,24,153,189,255,126
,255,36,102,254,170,254,0,0,0,0,0,28,28     CL
•87 DATA28,28,8,62,28,8,60,126,106,247,18
9,169,60,24,189,254,120,60,124,71,33,96     NM
•88 DATA222,127,30,60,62,226,132,6,56,126
,221,245,245,221,126,56,112,65,127,248      DO
•89 DATA248,127,65,112,0,0                  PL
•90 GOTO200                                 BO
•91 POKE53272,(PEEK(53272)AND240)+12:PRIN
T"[CLEAR]":FORT=1TO990:PRINT"[GREEN] ";:
NEXT                                        LC
•92 PRINT"[HOME][3"[DOWN]"][c 7][40"D"]";   KG
•93 PRINT"[5"[DOWN]"][40"D"]";              HK
•94 PRINT"[7"[DOWN]"][40"D"]";              EJ
•95 PRINT"[3"[DOWN]"][40"D"]";              GB
•96 POKEVV+23,25:POKEVV+29,13:POKEVV+27,2
55:POKEVV,18:POKEVV+1,208:POKEVV+4,125      FG
•97 POKEVV+5,229:POKEVV+2,80:POKEVV+3,229
:POKEVV+6,90:POKEVV+7,208                   CI
•98 POKEVV+8,60:POKEVV+9,208:POKEVV+10,16
0:POKEVV+11,229:POKEVV+21,255:GOTO30        JN
•99 STOP                                    JC
•100 POKE56334,PEEK(56334)AND254:POKE1,PE
EK(1)AND251                                 BE
•110 FORI=0TO511:POKEI+12288,PEEK(I+53248
):NEXT                                      JI
•120 POKE1,PEEK(1)OR4                       BE
•130 POKE56334,PEEK(56334)OR1               PD
•140 GOTO81                                 PP
•150 POKES+1,0:POKES,0:POKEA,32:POKEA+40,
9:POKEA+41,10:FORT=1TO50:PRINT:NEXT         KF
•155 X=54272                                BJ
•157 POKEVV+27,.                            LL
•160 POKE53272,21:FORQW=1944TO1184STEP-78
:POKEQW,8:POKEQW+X,0:POKEQW+1,1             NE
•165 POKEQW+1+X,0:FORSR=1TO400:NEXT:NEXT:
POKEVV+21,0:GOTO58                          JK
•200 REM SPRITES!!!!                        EA
•210 VV=53248:POKE2040,209:POKE2041,209:P
OKE2042,209:POKE2043,209                    LD
•212 POKE2044,209:POKE2045,209              GE
•213 POKEVV+43,8:POKEVV+44,10               NI
•215 POKEVV+39,10:POKEVV+40,10:POKEVV+41,
2:POKEVV+42,8                               BL
```

```
•220 POKE2046,196:FORTT=13376TO13438:READ
 T:POKETT,T:NEXT                          JG
•300 DATA0,0,0,0,0,0,0,16,0,0,56,0,0,120,
 0,0,124,0,0,254,0,1,255,0,7,119,0,15,175 JG
•310 DATA128,31,223,128,27,254,192,21,253
 ,240,30,251,248,63,127,120,63,188,252    AB
•320 DATA127,225,252,252,255,252,219,255,
 254,231,255,255,255,255,255              BL
•500 GOTO20                               OK
•600 Y=1                                  DJ
•601 GETA$:IFA$=""THEN601                 IE
•610 IFA$="[F1]"THEN620                   LF
•611 GOTO91                               PO
•620 Y=Y+1:IFY>5THENY=1                   DE
•630 PRINT"[HOME]LEVEL ";Y:GOTO601        HA
```

VIC 20 VERSION

```
•2 GOTO80                                 PA
•3 E=+1:POKEA,32:POKEA+22,32::A=A+E:IFPEE
 K(A+44)=32THEN7                          CM
•4 GOTO37                                 PC
•5 E=-1:POKEA,32:POKEA+22,32:A=A+E:IFPEEK
 (A+44)=32THEN7                           DH
•6 GOTO37                                 PC
•7 L=.:T=241:POKEV-2,.                    IC
•8 POKEA,32:POKEA+22,32:L=L+1:A=A+22:IFPE
 EK(A+44)=4THEN12                         FG
•9 IFA=BTHEN64                            NK
•10 POKEA,.:POKEA+22,1:POKEU,T:T=T-3:GOTO
 8                                        BM
•12 IFL>=9THEN55                          HP
•13 POKEU,.:GOTO37                        HA
•14 Y=135:BJ=BJ+1                         JB
•15 POKEV-2,Y:Y=Y+5:POKEA,32:POKEA+22,32:
 IFPEEK(A-22)=4THENA=A+1:GOTO7            BF
•17 IFA<7702THENA=A+1:GOTO7               DG
•18 IFA-22=NORPEEK(A-22)=5THEN55          HP
•19 A=A-22:POKEA,.:POKEA+22,1:GOTO15      LO
•20 POKE36869,255:PRINT"[CLEAR][DOWN][DOW
 N][5"[RIGHT]"][RED][RVSON]HOP[RVSOFF] [R
 VSON]AROUND":PRINT"[GREEN][DOWN][8"[RIGH
 T]"][RVSON]BY[DOWN][LEFT][LEFT]KEVIN[5"[
 LEFT]"][DOWN]DEWEY"                      FA
•21 PRINT"[BLUE][RVSON][DOWN][RIGHT]KEYS-
 LEFT=[PURPLE] :[BLUE][DOWN][7"[LEFT]"]RI
 GHT=[PURPLE] ;[BLUE][DOWN][8"[LEFT]"]JUM
 P=[PURPLE] J"                            DD
•22 PRINT"[DOWN][RED][RVSON]OBJECT:":PRIN
 T"[DOWN]@[DOWN][LEFT]A[UP] [RVSON]MUST S
 AVE[RVSOFF] F[DOWN][LEFT]G[UP] [RVSON]FR
 OM[RVSOFF] B[DOWN][LEFT]C,"              PK
•23 PRINT"[DOWN][RVSON]WHILE DODGING[RVSO
 FF] B[DOWN][LEFT]C[UP] [RVSON]AND[RVSOFF
 ] E!"                                    LN
•24 PRINT"[DOWN][DOWN][BLACK][RVSON]HIT A
 KEY TO START[3"."]"                      CK
•25 GETA$:IFA$=""THEN25                   EE
•26 GOTO89                                AH
•27 GU=1:POKEB-22,2:GOTO55                CN
•30 A=8109:B=7703:V=36878:POKEV,15:POKEV+
 1,24:U=36876:C=7744                      NB
•32 PRINT"[HOME][13"[DOWN]"][22"D"]";     JL
•34 POKEB,2:POKEB+22,3:D=PEEK(197):IFD=45
 THEN5                                    ON
•35 POKEC-22,6:POKEC,MZ:IFD=22THEN3       NL
•36 IFD=20THEN14                          PG
•37 POKEA,.:POKEA+22,1:IFPEEK(A+44)=32THE
 N7                                       BP
•38 M=INT(RND(1)*9)+1:IFM=9ANDP=.THEN42   CL
•39 IFA=BTHEN27                           ND
•40 IFP=1THENRETURN                       PH
•41 GOTO34                                PF
•42 FORT=241TO200STEP-1:POKEU,T:POKEU-1,T
 :NEXT:POKEU,.:N=INT(RND(1)*19)+7746:P=1:
 T=241                                    OD
•43 FORO=NTON+462STEP22:POKEN,5:POKEU,T:G
 OSUB34:POKEN,32:N=N+22:IFN=AORN=A+22THEN
 55                                       MA
•44 IFPEEK(N)=4THENPOKEV-1,135:FORT=1TO10
 :NEXT:POKEV-1,.:N=N+1                    LF
•45 NEXT:POKEU,.                          AM
•46 P=0:POKEN,32:N=.:POKEB,32:POKEB+22,32
 :B=B+1:IFPEEK(B)=6THEN49                 LL
•47 IFMZ=8THENMZ=7:GOTO34                 LF
•48 IFMZ=7THENMZ=8:GOTO34                 JP
•49 POKEV-2,.:TR=240                      GE
•50 FORT=7744TO8186STEP22:POKET,7:POKET-2
 2,6:POKET-1,3:POKET-23,2:POKEV-1,TR:TR=T
 R-5                                      CC
•51 FORTT=1TO20:NEXT:POKET,32:POKET-22,32
 :POKET-1,32:POKET-23,32:NEXT             BM
•52 POKEV-1,.:FORTT=1TO300:NEXT           HI
•55 RR=2:POKEU,.:POKEA,32:POKEA+22,9:POKE
 A+23,10:POKEU+1,200:IFGU=1THENPOKEB,3:GU
 =.                                       MI
•56 FORT=1TO450:NEXT:POKEU+1,.:FORT=1TO50
 0:NEXT                                   OC
•57 POKEV,.                               EK
•58 PRINT"[CLEAR][5"[DOWN]"][3"[RIGHT]"]Y
 OU BOMBED OUT.":POKE36869,240            HM
•59 PRINT"[3"[DOWN]"]HIT A KEY TO TRY AGA
 IN":PRINT"[4"[DOWN]"](DO IT RIGHT[10" "]
 THIS TIME!)"                             JJ
•61 IFPEEK(197)<>64ANDPEEK(197)<>39THENRU
 N90                                      HI
•62 GOSUB96                               AF
•63 GOTO61                                PF
•64 POKEB-1,.:POKEB+21,1                  MA
•65 Y=135:FORT=BTO8184STEP+23:POKET,2:POK
 ET+22,3:POKEV-2,Y:Y=Y+5:FORTT=1TO15:NEXT HO
•67 POKET,32:POKET+22,32:NEXT:POKEV-2,0:F
 ORT=1TO500:POKEV-1,135:NEXT:POKEV-1,.    LA
•69 POKEV,.:RR=1:POKE36869,240:PRINT"[CLE
 AR][5"[DOWN]"][5"[RIGHT]"][RVSON]YOU DID
 IT!"                                     IF
•70 PRINT"[DOWN]YOU SAVED SWEET DORIS.":P
```

```
RINT"[DOWN][DOWN]WITH ONLY";BJ;" JUMPS!" EE
·71 PRINT"[DOWN][DOWN]DO YOU WISH TO PLAY
[3" "]AGAIN,HERO?"                       CN
·72 PRINT"[DOWN][DOWN]IF SO,HIT ANY KEY." BD
·73 IFPEEK(197)<>64ANDPEEK(197)<>39THENRU
N90                                      HI
·74 GOSUB96                              AF
·75 GOTO73                               PC
·80 POKE52,28:POKE56,28:CLR:FORI=7168TO76
79:POKEI,PEEK(I+25600):NEXT:FORA=7168TO7
257                                      EH
·81 READB:POKEA,B:NEXT                   BN
·83 DATA60,126,219,255,231,126,66,60,24,2
55,189,189,60,36,36,102,126,90,126,255,1
29,195                                   FJ
·84 DATA102,60,24,153,189,255,126,255,36,
102,254,170,254,0,0,0,0,0,28,28,28,28,8,
62,28,8                                  FK
·85 DATA60,126,106,247,189,164,60,24,189,
254,120,60,124,71,33,96,222,127,30,60,62
,226,132                                 LK
·86 DATA6,56,126,221,245,245,221,126,56,1
12,65,127,248,248,127,65,112,0,0         FJ
·87 GOTO20                               OK
·89 GT=0:POKE646,GT                      LE
·90 PRINT"[CLEAR]";:POKE36869,255:FORT=1T
O490:PRINT" ";:NEXT                      KJ
·91 PRINT"[HOME][3"[DOWN]"][22"D"][5"[DOW
N]"][22"D"]";                            KM
·92 PRINT"[7"[DOWN]"][22"D"][3"[DOWN]"][2
2"D"]";                                  AK
·93 MZ=7:GOTO30                          LD
·96 IFPEEK(197)=39THENGT=GT+1:POKE646,GT:
IFRR=2THEN58                             AJ
·97 IFPEEK(197)=39ANDRR=1THEN69          IB
·98 IFGT>6THENGT=0                       DD
·99 POKE646,GT:RETURN                    NO
```

Space Hunt

FROM PAGE 43

```
·10 REM ANTHONY WOOD                     OI
·20 REM 12739 GRAND CROSS LN.            HI
·30 REM HOUSTON, TX 77072                MJ
·100 REM 6/18/84                         AJ
·110 PRINTCHR$(142);"[CLEAR][WHITE][DOWN]
[DOWN][6" "][s U][s C][s C][s U][s C][s
I][s U][s C][s I][s U][s C][s I][c A][s
*]  [c R] [c R][c A] [c S][s U][s I][c R
][c A][c R][c S]"                        NL
·120 PRINT"[6" "][s J][s C][s I][c Q][s C
][s K][c Q][s C][c W][s B]  [c Q][s C]
[c Q][s C][c W][s B] [4"[s B]"] [s B] "  KH
·130 PRINT"[6" "[3"[s C]"][s K][c E]  [c
E] [c E][s J][s C][s K][c Z][s C]  [c E]
 [c E][s J][s C][s K][c E][s J][c X] [c
E] "                                     PM
·140 PRINT"[DOWN][9" "](C) COPYRIGHT 1984
"                                        DG
·150 PRINT"[12" "]ANTHONY WOOD"          DF
·160 DIMD$(13),N(13),S(13),E(13),W(13),I(
13,10),IN(10),N$(20),T(12):NC=0          DO
·170 ER$="[s I] DON'T SEE ONE HERE!":UN$=
"[s I] DON'T UNDERSTAND."                IF
·180 CG$="[s I] CAN'T GET THAT!"         HH
·190 FORX=0TO13:READD$(X),N(X),S(X),E(X),
W(X),K:IFK=0THEN 210                     GJ
·200 FORY=0TOK-1:READA$:GOSUB 2240:I(X,Y)
=NC:NEXT                                 KM
·210 NEXT                                IA
·220 L=0:I(9,0)=7                        IB
·230 PRINT"[16"[DOWN]"]PRESS RETURN TO ST
ART.";                                   MH
·240 GETA$:IFA$<>CHR$(13)THEN 240        DL
·250 PRINT"[CLEAR]";CHR$(14);            EK
·260 PRINT"[CLEAR][s I] AM ";D$(L):PRINT DM
·270 IFN(L)+S(L)+E(L)+W(L)=-4THEN 340    BE
·280 PRINT"EXITS: ";                     BH
·290 IFN(L)<>-1THENPRINT"NORTH. ";       EB
·300 IFS(L)<>-1THENPRINT"SOUTH. ";       HI
·310 IFE(L)<>-1THENPRINT"EAST. ";        CD
·320 IFW(L)<>-1THENPRINT"WEST.";         AC
·330 PRINT                               JJ
·340 S=0:FORX=0TO10:S=S+I(L,X):NEXT:IFS=0
THEN 380                                 GD
·350 PRINT:PRINT"[s I] CAN SEE:"         NF
·360 FORX=0TO10:IFI(L,X)<>.THENPRINTN$(I(
L,X));". ";                              MK
·370 NEXT:PRINT                          FO
·380 PRINT"[40"[s *]"]"                  LA
·390 PRINT"[s W]HAT NOW? ";:GOSUB 2020   AL
·400 IFI$=""THEN 390                     CJ
·410 IFI$="[CLEAR]"THEN 260              ML
·420 IFV$<>"I"ANDLEFT$(V$,3)<>"INV"THEN 4
60                                       MG
·430 S=0:FORX=0TO10:S=S+IN(X):NEXT:IFS=0T
HENPRINT"[s I]'M NOT CARRYING ANYTHING."
:GOTO 390                                PH
·440 FORX=0TO10:IFIN(X)=0THENNEXT:GOTO 39
0                                        BC
·450 PRINTN$(IN(X)):NEXT:GOTO 390        JN
·460 IFV$="N"ORI$="NORTH"THENEX=N(L):GOTO
 510                                     IE
·470 IFV$="S"ORI$="SOUTH"THENEX=S(L):GOTO
 510                                     IA
·480 IFV$="E"ORI$="EAST"THENEX=E(L):GOTO
510                                      CL
·490 IFV$<>"W"ANDI$<>"WEST"THEN 530      FJ
·500 EX=W(L)                             EA
·510 IFEX=-1THENPRINT"[s I] CAN'T GO THAT
 WAY!":GOTO 390                          DO
·520 L=EX:LD=0:GOTO 260                  NO
```

```
·530 L$=LEFT$(N$,3):IFL$="TRA"THENL$="PIL
 ":GOTO 650                              DO
·540 IFL$="LOC"THENL$="EQU":GOTO 650     LA
·550 IFL$="TRU"THENL$="WOO":GOTO 650     FF
·560 IFL$="SUI"THENL$="PRE":GOTO 650     DD
·570 IFL$="SHI"THENL$="SPA":GOTO 650     DD
·580 IFL$="BUT"THENL$="RED":GOTO 650     AP
·590 IFL$="EQU"THENL$="LOT":GOTO 650     DF
·600 IFL$="BOA"THENL$="CON":GOTO 650     EB
·610 IFL$="TRE"THENL$="FAB":GOTO 650     EL
·620 IFL$="BOU"THENL$="LAR":GOTO 650     IO
·630 IFL$="STA"THENL$="AIR":GOTO 650     OL
·640 IFL$="DOO"AND(L=9ORL=7)THENL$="SPA" MF
·650 V$=LEFT$(V$,3):IFL$="321"ORL$="DOW"T
 HENNN=99:GOTO 710                       MA
·660 IFV$="LAY"ORV$="LIE"ORV$="SAV"ORV$="
 LOA"THEN 710                            HP
·670 IFV$="LOO"THEN 260                  EC
·680 IFV$="QUI"THENPRINT"[CLEAR][DOWN]TYP
 E 'GOTO 250' TO RECOVER.":END           ON
·690 FORNN=1TONC:IFL$=LEFT$(N$(NN),3)THEN
  710                                    KL
·700 NEXT:PRINTUN$:GOTO 390              JK
·710 IFV$<>"EXA"ANDV$<>"SEA"THEN 910     DL
·720 GOSUB 2190:IFF=0THENPRINT"[s I] DON'
 T SEE A ";N$;" HERE!":GOTO 390          NH
·730 IFL$="EQU"THENPRINT"[s I]T'S A METAL
  STORAGE LOCKER.":GOTO 390              IF
·740 IFL$="LAR"THENPRINT"[s T]HERE SEEMS
 TO BE SOMETHING UNDER HERE.":GOTO 390   OI
·750 IFL$="CON"THENPRINT"[s I] SEE A CALC
 ULATOR TYPE KEYBOARD HERE.":GOTO 390    EF
·760 IFL$<>"PRE"THEN 800                 NH
·770 IFCC=1THEN 900                      EK
·780 CC=1:PRINT"[s I] FOUND A SCREW DRIVE
 R HERE!":A$="SCREW DRIVER":GOSUB 2240   AD
·790 GOSUB 2270:GOTO 390                 KJ
·800 IFL$="SPA"THENPRINT"[s I] CAN SEE A
 DOOR.":GOTO 390                         PC
·810 IFL$<>"WOO"THEN 840                 ND
·820 PRINT"[s I]T'S A WOODEN TRUNK.":IFLK
 =0THENPRINT"[s I]T'S LOCKED."           NM
·830 GOTO 390                            CE
·840 IFL$<>"PIL"THEN 860                 LD
·850 IFKF=0THENPRINT"[s I] FIND A KEY HER
 E!":A$="KEY":GOSUB 2240:GOSUB 2270:KF=1:
 GOTO 390                                FO
·860 IFL$<>"LOT"THEN 900                 OM
·870 IFPP=1THEN 900                      FI
·880 PRINT"[s T]HERE IS AN ACCESS PANEL S
 CREWED DOWN  HERE.":A$="PANEL":GOSUB 224
 0                                       FE
·890 GOSUB 2270:PP=1:GOTO 390            GL
·900 PRINT"[s I] SEE NOTHING SPECIAL.":GO
 TO 390                                  JB
·910 IFV$<>"OPE"THEN 1040                AA
·920 GOSUB 2190:IFF=0THENPRINTER$:GOTO 39
 0                                       JP
·930 IFL$<>"EQU"THEN 990                 PL
·940 IFWS<>0THEN 970                     CK
·950 PRINT"[s A][s G][s G][s H][s H]!! [s
  I]'M ELECTROCUTED BY 10,[3"0"]":PRINT"V
 OLTS THAT WAS WIRED TO THE LOCKER."     ND
·960 END                                 IC
·970 IFNT=1THENPRINT"[s I]T'S EMPTY":GOTO
  390                                    BJ
·980 NT=1:PRINT"[s I] FIND A NOTE HERE!":
 A$="NOTE":GOSUB 2240:GOSUB 2270:GOTO 390 LC
·990 IFL$<>"WOO"THENPRINTUN$:GOTO 390    PH
·1000 IFLK=0THENPRINT"[s I]T'S LOCKED!":G
 OTO 390                                 IJ
·1010 IFTE=1THENPRINT"[s I]T'S EMPTY.":GO
 TO 390                                  NH
·1020 TE=1:PRINT"[s I] FOUND A PRESSURE S
 UIT IN HERE!":A$="PRESSURE SUIT":GOSUB 2
 240                                     FJ
·1030 GOSUB 2270:GOTO 390                KJ
·1040 IFV$<>"UNL"THEN 1090               BO
·1050 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
·1060 IFL$<>"WOO"THENPRINTUN$:GOTO 390   PH
·1070 A$="KEY":GOSUB 2430:GOSUB 2190:IFF=
 0THENPRINT"[s I] NEED A KEY.":GOTO 390  KG
·1080 LK=1:PRINT"OK":GOTO 390            NH
·1090 IFV$<>"WEA"THEN 1140               BO
·1100 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
·1110 IFL$<>"PRE"THENPRINTUN$:GOTO 390   PH
·1120 IFF=1THENGOSUB 2310:GOSUB 2350     EI
·1130 PRINT"OK":WS=1:GOTO 390            LA
·1140 IFV$<>"REA"THEN 1180               OH
·1150 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
·1160 IFL$<>"NOT"THENPRINTUN$:GOTO 390   DF
·1170 PRINT"[s I]T SAYS '3212'":GOTO 390 CL
·1180 IFV$<>"PRE"ANDV$<>"PUS"THEN 1240   FD
·1190 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
·1200 IFL$<>"RED"THENPRINTUN$:GOTO 390   PH
·1210 IFWS=0THENPRINT"[s A][s G][s G][s H
 ][s H][3"!"] [s T]HE AIRLOCK CYCLES OUT.
  [s I]'M[4" "]DEAD!":END                LC
·1220 IFW(L)=0THENW(L)=-1:E(L)=7:GOTO 260 JJ
·1230 W(L)=0:E(L)=-1:GOTO 260            CK
·1240 IFV$<>"GO"ANDV$<>"ENT"THEN 1300    DP
·1245 IFL$="321"THEN 1430                PI
·1250 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
·1260 IFL$="AIR"THENL=4:GOTO 260         FJ
·1270 IFL$="SPA"THENL=6:GOTO 260         FF
·1280 IFL$="TUN"THENL=12:GOTO 260        GG
·1290 PRINTUN$:GOTO 390                  MB
·1300 IFV$<>"GET"ANDV$<>"TAK"ANDV$<>"PIC"
 THEN 1370                               AM
·1310 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
 90                                      JP
```

```
•1320 IFF=2THENPRINT"[s I]'M ALREADY CARR
YING IT!":GOTO 390                        EM
•1330 IFL$="EQU"ORL$="PIL"ORL$="LOT"ORL$=
"TUN"ORL$="CON"THENPRINTCG$:GOTO 390      FL
•1340 IFL$="SPA"ORL$="LAR"ORL$="AIR"THENP
RINTCG$:GOTO 390                          EH
•1350 IFL$="FAB"THENPRINT"** [s C]ONGRATU
LATIONS **[3" "][s Y]OU HAVE FOUND  THE
TREASURE!":END                            KH
•1360 GOSUB 2350:GOSUB 2310:PRINT"OK":GOT
0 390                                     NO
•1370 IFV$<>"DRO"ANDV$<>"PUT"THEN 1420    BK
•1380 GOSUB 2190:IFF<>2THENPRINTER$:GOTO
390                                       DK
•1390 IFL$="PRE"ANDL>6THENPRINT"[s A][s U
][s G][s G][s H][s H]!! [s I]'M SPACED!"
:END                                      BE
•1400 IFL$="PRE"THENWS=0                  AH
•1410 GOSUB 2390:GOSUB 2270:PRINT"OK":GOT
0 390                                     BD
•1420 IFV$<>"TYP"THEN 1470                AI
•1430 IFL<>6THENPRINT"[s I] CAN'T DO THAT
 HERE!":GOTO 390                          OL
•1440 IFL$<>"321"THENPRINT"[s N]OTHING HA
PPENS.":GOTO 390                          CM
•1450 IFS(L)=9THENPRINT"[s N]OTHING HAPPE
NS.":GOTO 390                             ED
•1460 S(L)=9:PRINT"[s T]HE SHIP TAKES OFF
, AND WE SOON LAND[4" "]ELSEWHERE.":GOTO
 390                                      HJ
•1470 IFV$<>"UNS"THEN 1540                AH
•1480 IFL<>5THENPRINT"[s I] CAN'T DO THAT
 HERE!":GOTO 390                          PG
•1490 A$="SCREW DRIVER":GOSUB 2430:GOSUB
2190:IFF<>0THEN 1510                      FP
•1500 PRINT"[s I] NEED A SCREW DRIVER!":G
OTO 390                                   MK
•1510 IFPA=1THENPRINT"[s I]T'S ALREADY UN
SCREWED.":GOTO 390                        GH
•1520 PA=1:PRINT"OK":PRINT"[s I] FOUND A
BLASTER HERE!":A$="BLASTER":GOSUB 2240    CH
•1530 GOSUB 2270:GOTO 390                 KJ
•1540 IFV$<>"BLA"ANDV$<>"SHO"THEN 1610    PO
•1550 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
90                                        JP
•1555 Z=NN:A$="BLASTER":GOSUB2430:GOSUB21
90:IFF=0THENPRINT"[s I] NEED A ";A$:GOTO
 390                                      PD
•1560 IFL$="BLA"THENPRINT"AT WHAT?":GOTO
390                                       BE
•1570 IFL$<>"LAR"THENPRINT"[s I] CAN'T SH
OOT THAT!":GOTO 390                       AK
•1580 IFLD=0THENPRINT"[s A][s G][s G][s H
][s H]! [s S]TANDING UP LIKE THIS, I'M H
IT[3" "]BY ROCKS!":END                    AI
•1590 PRINT"[s K][s A][s A][s B][s O][s O
][s M]!!  [s A] TUNNEL WAS UNDER THE[7"
"]BOULDER!":NN=Z                          GB
```

```
•1600 GOSUB2310:A$="TUNNEL":GOSUB 2240:GO
SUB 2270:GOTO 390                         HI
•1610 IFV$<>"FIR"THEN 1650                BI
•1620 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
90                                        JP
•1630 IFL$<>"BLA"THENPRINTUN$:GOTO 390    NL
•1640 PRINT"TRY SHOOT [4"."]":GOTO 390    IC
•1650 IFV$<>"LAY"ANDV$<>"LIE"THEN 1680    IC
•1660 IFL$<>"DOW"ANDL$<>""THENPRINTUN$:GO
TO 390                                    EN
•1670 PRINT"OK":LD=1:GOTO 390             LM
•1680 IFV$<>"SAV"THEN 1780                CF
•1690 F$=N$+".ADV":OPEN15,8,15,"I0":PRINT
#15,"S0:"+F$:OPEN2,8,2,F$+",S,W"          NC
•1700 INPUT#15,E,ER$,Z,Z:IFE<>0THENPRINTE
R$:CLOSE2:CLOSE15:GOTO 390                DD
•1710 PRINT#2,NC:FORX=1TONC:PRINT#2,N$(X)
:NEXT:FORX=0TO13:FORY=0TO10               MI
•1720 PRINT#2,CHR$(I(X,Y));:NEXT:NEXT     HM
•1730 FORX=0TO10:PRINT#2,CHR$(IN(X));:NEX
T:PRINT#2,CHR$(L);CHR$(CC);CHR$(PP);      CJ
•1740 PRINT#2,CHR$(NT);CHR$(TE);CHR$(LK);
CHR$(WS);CHR$(PA);CHR$(LD);CHR$(KF);      KN
•1750 IFW(4)=-1THENPRINT#2,CHR$(1);:GOTO
1770                                      NN
•1760 PRINT#2,CHR$(0);                    IK
•1770 PRINT#2,CHR$(S(6));:CLOSE2:CLOSE15:
PRINT"OK":GOTO 390                        AD
•1780 IFV$<>"LOA"THEN 1890                DB
•1790 F$=N$+".ADV":OPEN15,8,15,"I0":OPEN2
,8,2,F$+",S,R":INPUT#15,E,ER$,Z,Z         BD
•1800 IFE<>0THENPRINTER$:CLOSE2:CLOSE15:G
OTO 390                                   CM
•1810 INPUT#2,NC:FORX=1TONC:INPUT#2,N$(X)
:NEXT:FORX=0TO13:FORY=0TO10               PI
•1820 GET#2,A$:I(X,Y)=ASC(A$+CHR$(0)):NEX
T:NEXT:FORX=0TO10:GET#2,A$                DE
•1830 IN(X)=ASC(A$+CHR$(0)):NEXT:FORX=0TO
11:GET#2,A$:T(X)=ASC(A$+CHR$(0)):NEXT     MK
•1840 L=T(0)                              NF
•1850 CC=T(1):PP=T(2):NT=T(3):TE=T(4):LK=
T(5):WS=T(6):PA=T(7):LD=T(8):KF=T(9)      EJ
•1860 S(6)=T(11):IFT(10)=0THENE(4)=-1:W(4
)=0:GOTO 1880                             FK
•1870 E(4)=7:W(4)=-1                      EJ
•1880 CLOSE2:CLOSE15:GOTO 260             JO
•1890 IFV$<>"USE"THEN 1950                AJ
•1900 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
90                                        JP
•1910 IFL$="BLA"THENPRINT"[s T]RY SHOOT [
3"."]":GOTO 390                           OO
•1920 IFL$<>"KEY"THENPRINTUN$:GOTO 390    OD
•1930 A$="WOODEN TRUNK":GOSUB 2430:GOSUB
2190:IFF<>0THEN 1080                      OH
•1940 PRINT"[s I] CAN'T DO THAT HERE!":GO
TO 390                                    MC
•1950 IFV$<>"MOV"ANDV$<>"ROL"THENPRINTUN$
:GOTO 390                                 CL
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 93 and 94 explain these codes and provide other essential information on entering *Ahoy!* programs. Refer to these pages **before** entering any programs!

```
•1960 GOSUB 2190:IFF=0THENPRINTER$:GOTO 3
90                                        JP
•1970 IFL$<>"LAR"THENPRINT"[s I]'M NOT MA
KING MUCH PROGRESS.":GOTO 390              FP
•1980 PRINT"[s I]T'S TOO HEAVY.":GOTO 390 MH
•1990 REM                                  JD
•2000 REM INPUT ROUTINE                    EB
•2010 REM                                  JD
•2020 I$=""                                HD
•2030 PRINT"[cEP]";                        MK
•2040 GETA$:IFA$=""THEN 2040               KE
•2050 IFA$="[CLEAR]"THENI$=A$:RETURN       GP
•2060 PRINT"[LEFT] [LEFT]";A$;:IFA$=CHR$(
13)THEN 2090                               DJ
•2070 IFA$=CHR$(20)ANDI$<>""THENI$=LEFT$(
I$,LEN(I$)-1):GOTO 2030                    BG
•2080 I$=I$+A$::GOTO 2030                  GG
•2090 IFI$=""THENRETURN                    CG
•2100 FORS=1TOLEN(I$):IFMID$(I$,S,1)<>" "
THENNEXT:V$=I$:N$="":RETURN                MK
•2110 V$=LEFT$(I$,S-1):N$=RIGHT$(I$,LEN(I
$)-S)                                      PL
•2120 RETURN                               IM
•2130 REM                                  JD
•2140 REM SEARCH FOR NN                    NH
•2150 REM F=0 IF NOT FOUND                 EK
•2160 REM F=1 IF IN ROOM                   LD
•2170 REM F=2 IF IN INVENTORY              DM
•2180 REM                                  JD
•2190 FORX=0TO10:IFI(L,X)=NNTHENF=1:RETUR
N                                          LC
•2200 IFIN(X)=NNTHENF=2:RETURN             PH
•2210 NEXT:F=0:RETURN                      LL
•2220 REM                                  JD
•2230 REM ADD A$ TO NOUN LIST. NN=POS      AO
•2240 NC=NC+1:N$(NC)=A$:NN=NC:RETURN       KJ
•2250 REM                                  JD
•2260 REM ADD NN TO ROOM                   ND
•2270 FORX=0TO10:IFI(L,X)=0THENI(L,X)=NN:
RETURN                                     DM
•2280 NEXT:PRINT"ERROR!":RETURN            AJ
•2290 REM                                  JD
•2300 REM DELETE NN FROM ROOM              FJ
•2310 FORX=0TO10:IFI(L,X)=NNTHENI(L,X)=0:
RETURN                                     KK
•2320 NEXT:PRINT"ERROR!":RETURN            AJ
•2330 REM                                  JD
•2340 REM ADD NN TO INVENTORY              FJ
•2350 FORX=0TO10:IFIN(X)=0THENIN(X)=NN:RE
TURN                                       MN
•2360 NEXT:PRINT"ERROR!":RETURN            AJ
•2370 REM                                  JD
•2380 REM DELETE NN FROM INVE              FA
•2390 FORX=0TO10:IFIN(X)=NNTHENIN(X)=0:RE
TURN                                       CD
•2400 NEXT:PRINT"ERROR!":RETURN            AJ
•2410 REM                                  JD
•2420 REM SET NN TO NO. FOR A$             OI
•2430 FORNN=1TONC:IFN$(NN)=A$THENRETURN    MB
•2440 NEXT:NN=99:RETURN                    FC
•2450 DATA "IN THE ENTRANCE CHAMBER OF A
LUNAR[HOME][DOWN]SPACE STATION.",-1,1,4,
-1,1                                       NM
•2460 DATA "EQUIPMENT LOCKER"              EJ
•2470 DATA "IN A NARROW HALLWAY THAT RUNS
[HOME][DOWN]NORTH-SOUTH.",0,3,2,-1,0       JC
•2480 DATA "IN A LONG DESERTED BUNK ROOM.
",-1,-1,-1,1,1,"WOODEN TRUNK"              EP
•2490 DATA "IN A RECTANGULAR ROOM THAT SM
ELLS[HOME][DOWN]MUSTY AND OLD.",1,-1,-1,
-1,1                                       HA
•2500 DATA "PILE OF TRASH"                 IL
•2510 DATA "IN A SPACE STATION AIRLOCK.",
-1,-1,-1,0,1,"RED BUTTON"                  NK
•2520 DATA "IN A SPACE SHIP'S ENGINE ROOM
.",-1,-1,6,-1,1                            MO
•2530 DATA"LOTS OF COMPLICATED EQUIPMENT" FF
•2540 DATA "IN A SPACE SHIP CONTROL ROOM.
",-1,7,-1,5,1,"CONTROL BOARD"              KK
•2550 DATA "ON AN OPEN LUNAR PLAIN.",-1,-
1,-1,-1,2,"SPACE SHIP"                     OC
•2560 DATA "AIRLOCK TO SPACE STATION"      LD
•2570 DATA "IN AN ASTEROID CRATER.",-1,9,
-1,11,0                                    EL
•2580 DATA "ON AN ASTEROID PLAIN.",8,-1,-
1,10,0                                     KO
•2590 DATA "ON AN ASTEROID PLAIN.",11,-1,
9,-1,1,"LARGE BOULDER"                     AI
•2600 DATA "IN THE HILLS ON AN ASTEROID."
,-1,10,8,-1,0                              CH
•2610 DATA "IN A SLOPING TUNNEL.",10,13,-
1,-1,0                                     GH
•2620 DATA "IN THE UNDERGROUND TREASURE R
OOM.",-1,-1,-1,-1,1                        MB
•2630 DATA "FABULOUS TREASURE"             BL
```

BOOTER

FROM PAGE 41 **C-64 VERSION**

```
•0 REM 'C64 BOOTER' COPYRIGHT 1984 GEORGE
 JONES * FOR C-64                          EC
•1 CLR:DV=PEEK(186)                        LJ
•2 BS%=692:FI%=512:CS%=93:SR%=735:RN%=733
:LM%=702:PO%=723:IM%=770:MX=65535          CJ
•3 S%=679:E%=771:POKE808,234:REM DISABLE
RUN/STOP RESTORE                           IE
•4 GOSUB62:REM TITLE                       EA
•5 GOSUB18:REM INPUT DISK,ID & PROG NAME
STRING                                     DK
•6 GOSUB47:REM FORMAT DISK                 OG
•7 PRINT"[CLEAR]":PRINT"[RIGHT][5"[DOWN]"
][BLACK][s T]HE [s B][s O][s O][s T] WIL
```

```
L LOAD AND RUN THE PROGRAM"                    EO
•8 PRINT"[RIGHT][DOWN][RED][RVSON]";PN$;:
PRINT"[RVSOFF][BLACK] AFTER YOU HAVE SAV
ED IT"                                         BK
•9 PRINT"[RIGHT][DOWN]ONTO THIS DISK.[DOW
N][RIGHT]"                                     EA
•10 PRINTTAB(6)" [s T]O [s R][s U][s N],
TYPE [s L][s O][s A][s D]";:PRINTCHR$(34
)":*"CHR$(34);:PRINT",8,1"                     LK
•11 CLOSE2:CLOSE15                             BI
•12 PRINT"[DOWN][10"[RIGHT]"][s A]NOTHER
[s B][s O][s O][s T] ? (Y/N)"                  HL
•13 GETA$:IFA$=""THEN13                        EH
•14 IFA$<>"Y"ANDA$<>"N"THEN13                  EP
•15 IFA$="Y"THENRUN                            IK
•16 PRINT"[CLEAR]":SYS58451:POKE808,237:R
EM RESTORE BASIC VECTORS & ENABLE RUN/ST
OP RESTORE                                     FH
•17 PRINT"[RVSON][RED][s A]LL [s D]ONE![D
OWN]";:GOTO17                                  HM
•18 PRINT"[CLEAR]":DN$="":ID$="":PN$="":P
OKERN%,174:POKERN%+1,167:POKEIM%,180:POK
EIM%+1,2                                       CN
•19 REM POKE ADDRESS OF BOOT, NORMALIZE B
ASIC WARM START VECTOR                         LF
•20 PRINTTAB(3)"[RED][RVSON][s P][s L][s
E][s A][s S][s E][SS][s I][s N][s S][s E
][s R][s T][SS][s D][s I][s S][s K][SS][
s T][s O][SS][s B][s E][SS][s F][s O][s
R][s M][s A][s T][s T][s E][s D][RVSOFF]
":PRINT                                        KN
•21 FORC=S%TOE%:READZ%:POKEC,Z%:NEXT           MC
•22 PRINTTAB(1)"** [RVSON][s A][s L][s L]
[SS][s D][s A][s T][s A][SS][s O][s N][S
S][s D][s I][s S][s K][SS][s W][s I][s L
][s L][SS][s B][s E][SS][s E][s R][s A][
s S][s E][s D]![RVSOFF] **"                    LD
•23 PRINT"[DOWN][DOWN]";:INPUT"[BLUE][RVS
ON][s N][s E][s W][SS][s D][s I][s S][s
K][SS][s N][s A][s M][s E]:[RVSOFF] ";DN
$                                              AN
•24 IFLEN(DN$)=0THEN23                         FI
•25 PRINT"[DOWN]";:INPUT"[BLUE][RVSON][s
N][s E][s W][SS][s D][s I][s S][s K][SS]
[s I][s D]:[RVSOFF] ";ID$                      MN
•26 IFLEN(ID$)=0THEN25                         FD
•27 PRINT"[DOWN][RIGHT][s N]AME THE PROGR
AM YOU WANT TO [s B]OOT"                       LN
•28 INPUTPN$                                   EO
•29 IFLEN(PN$)=0THEN28                         FL
•30 GOSUB67:REM GET STRING FROM INPUT BUF
FER AND POKE TO FILENAM                        PN
•31 PRINT"[DOWN][s I]S [RVSON]";PN$;"[RVS
OFF] A [s B][s A][s S][s I][s C] PROGRAM
? (Y/N)"                                       LD
•32 GETA$:IFA$=""THEN32                        EG
•33 IFA$<>"Y"ANDA$<>"N"THEN32                  EK
•34 IFA$="Y"THENRETURN:REM NOW GET ENTRY
POINT OF ML PROGRAM                            JF
•35 INPUT"[DOWN][RVSON][s W]HAT IS THE [s
 S][s Y][s S] ADDRESS:[RVSOFF]";AD             BM
•36 IFAD=<1ORAD>=MXTHENPRINT"[DOWN][DOWN]
[4"[RIGHT]"][RVSON][RED][s O][s U][s T]
[s O][s F] [s R][s A][s N][s G][s E]![RV
SOFF][BLUE]":FORC=1TO1000:NEXT:PRINT"[DO
WN][DOWN]";:GOTO35                             OJ
•37 POKERN%,(AD-INT(AD/256)*256):REM POKE
 LO BYTE                                       OJ
•38 POKERN%+1,(AD/256):REM POKE HI BYTE        BH
•39 PRINTTAB(4)"[DOWN][s A]RE YOUR ENTRIE
S CORRECT? (Y/N)"                              LK
•40 GETA$:IFA$=""THEN40                        EH
•41 IFA$<>"Y"ANDA$<>"N"THEN40                  EL
•42 IFA$="N"THENRUN                            FP
•43 RETURN                                     IM
•44 CLOSE15:OPEN15,DV,15:REM ERROR CHECK       BK
•45 INPUT#15,ER,ER$,T,S                        LB
•46 RETURN                                     IM
•47 GOSUB44:REM ERROR CHECK                    MF
•48 IFERTHENPRINTTAB(8)ER;ER$;T;S:PRINT"[
RVSON][RED][s P][s L][s E][s A][s S][s E
] [s P][s O][s W][s E][s R] [s D][s O][s
 W][s N][SS]& [s S][s T][s A][s R][s T]
[s O][s V][s E][s R]!";:GOTO48                 CP
•49 CLOSE15:OPEN15,DV,15                       KN
•50 PRINT#15,"N0:"+DN$+","+ID$:REM TITLE
NEW DISK                                       BC
•51 CLOSE15                                    AB
•52 GOSUB44                                    PM
•53 IFERTHEN48                                 DP
•54 CLOSE2:OPEN2,DV,1,"0:[SS]THE MAGIC BO
OT,P,W":REM SHIFTED SPACE BEFORE TITLE         CD
•55 PRINT#2,CHR$(S%-INT(S%/256)*256);          NI
•56 PRINT#2,CHR$(S%/256);                      BF
•57 FORI=0TOCS%-1                              AD
•58 PRINT#2,CHR$(PEEK(S%+I));                  OH
•59 NEXT                                       IA
•60 CLOSE2                                     NC
•61 RETURN                                     IM
•62 PRINT"[CLEAR]";CHR$(14):POKE53280,1:P
OKE53281,1                                     BE
•63 PRINTTAB(15)"[3"[DOWN]"][BLUE][s C]64
 [s B][s O][s O][s T][s E][s R]"               II
•64 PRINTTAB(6)"[DOWN][s C]OPYRIGHT 1984
[s G]EORGE [s J]ONES"                          IB
•65 FORT=1TO2000:NEXT                          OB
•66 RETURN                                     IM
•67 FORC=1TOLEN(PN$)                           BK
•68 POKESR%,(PEEK(FI%))                        NL
•69 SR%=SR%+1:FI%=FI%+1                        IB
•70 NEXT                                       IA
•71 POKELM%,LEN(PN$):REM SET LENGTH OF NE
W PROGRAM NAME                                 EA
•72 RETURN                                     IM
•73 REM*******************************         KF
•74 REM  MX=65535 MAXIMUM ADDRESS              IK
```

```
•75 REM  BS%=692 START OF NUBOOT         CH
•76 REM  FI%=512 SYSTEM INPUT BUFFER     DH
•77 REM  CS%=80 NO OF BYTES TO SAVE      KE
•78 REM  SR%=735 LOC OF FILE NAME        LD
•79 REM  RN%=733 INTERPRETER LOOP &      LP
•80 REM  SYS ADDRESS FOR ML PROG         DJ
•81 REM  LM%=702 STORE LEN(PN$)          NA
•82 REM  IM%=770 BASIC WARM START VECTOR EG
•83 REM  PN$= PROGRAM NAME               OA
•84 REM  DN$= DISK NAME                  MI
•85 REM  ID$= DISK ID                    FG
•86 REM****************************      GH
•87 DATA40,67,40,49,57,56,52,71,74,79,78,
 69,83,169,1,162                         ML
•88 DATA8,160,1,32,186,255,169,15,162,223
 ,160,2,32,189,255,169                   CO
•89 DATA0,162,255,160,255,32,213,255,134,
 45,132,46,32,239,2,32                   AO
•90 DATA83,228,32,89,166,76,174,167,160,1
 60,160,160,160,160,160,160              PJ
•91 DATA160,160,160,160,160,160,160,160,1
 62,0,169,160,157,223,2,232              OL
•92 DATA224,16,208,246,141,189,2,96,0,139
 ,227,180,2                              JB
•93 REM 'C64 BOOTER' COPYRIGHT 1984 GEORG
 E JONES * FOR C-64                      EC
```

VIC 20 VERSION

```
•0 REM 'VIC BOOTER' COPYRIGHT 1984 GEORGE
  JONES * FOR VIC 20                     CD
•1 CLR:DV=PEEK(186)                      LJ
•2 BS%=692:FI%=512:CS%=93:SR%=735:RN%=733
 :LM%=702:PO%=723:IM%=770:MX=65535       CJ
•3 S%=679:E%=771:LO%=PEEK(IM%):HI%=PEEK(E
 %)                                      LA
•4 GOSUB66:REM TITLE                     DA
•5 GOSUB19:REM INPUT DISK,ID & PROG NAME
 STRING                                  CL
•6 GOSUB51:REM FORMAT DISK AND WRITE BOOT MH
•7 PRINT"[CLEAR]";:PRINT"[3"[DOWN]"][BLAC
 K][s T]HE [s B][s O][s O][s T] WILL LOAD
 ":PRINT"[DOWN]AND RUN THE PROGRAM"      AK
•8 PRINT"[RIGHT][DOWN][RED][RVSON]";PN$:P
 RINT:PRINT"[RVSOFF][BLACK]AFTER YOU HAVE
  SAVED"                                 GF
•9 PRINT"[DOWN]IT ONTO THIS DISK.[DOWN][R
 IGHT]"                                  KG
•10 PRINT"[s T]O TEST, TYPE:[DOWN]":PRINT
 "[s L][s O][s A][s D]";:PRINTCHR$(34)":*
 "CHR$(34);:PRINT",8,1"                  FD
•11 CLOSE2:CLOSE15                       BI
•12 PRINT"[DOWN][RIGHT][s A]NOTHER [s B][
 s O][s O][s T] ? (Y/N)"                 DF
•13 GETA$:IFA$=""THEN13                  EH
•14 IFA$<>"Y"ANDA$<>"N"THEN13            EP
•15 IFA$="Y"THENRESTORE:GOTO5            IB
•16 PRINT"[CLEAR]"                       HH
•17 SYS58459:REM RESTORE WARM START VECTO
 R                                       AC
•18 PRINT"[RVSON][RED][s A]LL [s D]ONE![D
 OWN]";:GOTO18                           JK
•19 PRINT"[CLEAR]":DN$="":ID$="":PN$="":P
 OKERN%,174:POKERN%+1,199:POKEIM%,180:POK
 EIM%+1,2                                KG
•20 REM POKE ADDRESS OF BOOT INTO BASIC W
 ARM START VECTOR                        KG
•21 PRINT"[RED][RVSON][s I][s N][s S][s E
 ][s R][s T][SS][s D][s I][s S][s K][SS][
 s T][s O][SS][s F][s O][s R][s M][s A][s
  T][RVSOFF]":PRINT                      OE
•22 FORC=S%TOE%:READZ%:POKEC,Z%:NEXT     MC
•23 PRINT"[RVSON]*[s D][s A][s T][s A][SS
 ][s W][s I][s L][s L][SS][s B][s E][SS][
 s E][s R][s A][s S][s E][s D]*[RVSOFF]" DN
•24 PRINT"[DOWN][3" "][BLUE][RVSON][s N][
 s E][s W][SS][s D][s I][s S][s K][SS][s
 N][s A][s M][s E]:[RVSOFF] [DOWN]"      PB
•25 INPUTDN$                             EC
•26 IFLEN(DN$)=0THEN24                   FL
•27 PRINT"[DOWN][4" "][BLUE][RVSON][s N][
 s E][s W][SS][s D][s I][s S][s K][SS][s
 I][s D]:[RVSOFF] [DOWN]"                JK
•28 INPUTID$                             EB
•29 IFLEN(ID$)=0THEN27                   FF
•30 PRINT"[DOWN][s N]AME THE PROGRAM":PRI
 NT"[DOWN]YOU WANT TO [s B]OOT[DOWN]"    GK
•31 INPUTPN$                             EO
•32 IFLEN(PN$)=0THEN31                   FD
•33 GOSUB71:REM GET STRING FROM INPUT BUF
 FER AND POKE TO FILENAM                 DH
•34 PRINT"[DOWN][s I]S [RVSON]";PN$;"[RVS
 OFF]":PRINT"[DOWN]A [s B][s A][s S][s I]
 [s C] PROGRAM? (Y/N)"                   AC
•35 GETA$:IFA$=""THEN35                  EL
•36 IFA$<>"Y"ANDA$<>"N"THEN35            FH
•37 IFA$="Y"THENRETURN:REM NOW GET ENTRY
 POINT OF ML PROGRAM                     JB
•38 PRINT"[DOWN][RVSON][s G]IVE THE [s S]
 [s Y][s S] ADDRESS:[RVSOFF]"            KC
•39 INPUTAD                              BH
•40 IFAD=<1ORAD>=MXTHENPRINT"[DOWN][DOWN]
 [4"[RIGHT]"][RVSON][RED][s O][s U][s T]
 [s O][s F] [s R][s A][s N][s G][s E]![RV
 SOFF][BLUE]":FORC=1TO1000:NEXT:PRINT"[DO
 WN][DOWN]";:GOTO38                      AA
•41 POKERN%,(AD-INT(AD/256)*256):REM POKE
  LO BYTE                                MO
•42 POKERN%+1,(AD/256):REM POKE HI BYTE  BE
•43 PRINTTAB(4)"[DOWN][s I]S DATA CORRECT
 ? (Y/N)"                                NC
•44 GETA$:IFA$=""THEN44                  EL
•45 IFA$<>"Y"ANDA$<>"N"THEN44            FH
•46 IFA$="N"THENRUN                      FP
•47 RETURN                               IM
•48 CLOSE15:OPEN15,DV,15                 KN
•49 INPUT#15,ER,ER$,T,S                  LB
```

```
•50 RETURN                                  IM
•51 GOSUB48                                 PI
•52 IFERTHENPRINTTAB(8)ER;ER$;T;S:PRINT"[
 RVSON][RED][s P][s L][s E][s A][s S][s E
 ] [s P][s O][s W][s E][s R] [s D][s O][s
  W][s N][SS]& [s S][s T][s A][s R][s T]
 [s O][s V][s E][s R]!";:GOTO52             JN
•53 CLOSE15:OPEN15,DV,15                    KN
•54 PRINT#15,"N0:"+DN$+","+ID$              AB
•55 CLOSE15                                 AB
•56 GOSUB48                                 PI
•57 IFERTHEN52                              DG
•58 CLOSE2:OPEN2,DV,1,"0:[SS]THE MAGIC BO
 OT,P,W":REM SHIFTED SPACE BEFORE TITLE     JO
•59 PRINT#2,CHR$(S%-INT(S%/256)*256);       NI
•60 PRINT#2,CHR$(S%/256);                   BF
•61 FORI=0TOCS%-1                           AD
•62 PRINT#2,CHR$(PEEK(S%+I));               OH
•63 NEXT                                    IA
•64 CLOSE2                                  NC
•65 RETURN                                  IM
•66 PRINT"[CLEAR]";CHR$(14):POKE36879,25    KO
•67 PRINTTAB(7)"[3"[DOWN]"][BLUE][s V][s
 I][s C] [s B][s O][s O][s T][s E][s R][D
 OWN]"                                      LN
•68 PRINT" ([s C]) 1984 [s G]EORGE [s J]O
 NES"                                       OI
•69 FORT=1TO2000:NEXT                       OB
•70 RETURN                                  IM
•71 FORC=1TOLEN(PN$)                        BK
•72 POKESR%,(PEEK(FI%))                     NL
•73 SR%=SR%+1:FI%=FI%+1                     IB
•74 NEXT                                    IA
•75 POKELM%,LEN(PN$):REM SET LENGTH OF NE
 W PROGRAM NAME                             IK
•76 RETURN                                  IM
•77 REM******************************       KF
•78 REM  MX=65535 MAXIMUM ADDRESS           IB
•79 REM  BS%=692 START OF NUBOOT            EN
•80 REM  FI%=512 SYSTEM INPUT BUFFER        OG
•81 REM  CS%=80 NO OF BYTES TO SAVE         BK
•82 REM  SR%=735 LOC OF FILE NAME           AI
•83 REM  RN%=733 INTERPRETER LOOP &         GC
•84 REM  SYS ADDRESS FOR ML PROG            BI
•85 REM  LM%=702 STORE LEN(PN$)             IL
•86 REM  IM%=770 BASIC WARM START VECTOR    PC
•87 REM  PN$= PROGRAM NAME                  BJ
•88 REM  DN$= DISK NAME                     MM
•89 REM  ID$= DISK ID                       CC
•90 REM******************************       GH
•91 DATA40,67,40,49,57,56,52,71,74,79,78,
 69,83,169,1,162                            ML
•92 DATA8,160,1,32,186,255,169,15,162,223
 ,160,2,32,189,255,169                      CO
•93 DATA0,162,255,160,255,32,213,255,134,
 45,132,46,32,239,2,32                      AO
•94 DATA91,228,32,89,198,76,174,199,160,1
 60,160,160,160,160,160,160                 GI
•95 DATA160,160,160,160,160,160,160,160,1
 62,0,169,160,157,223,2,232                 OL
•96 DATA224,16,208,246,141,189,2,96,0,139
 ,227,180,2                                 JB
•97 REM 'VIC BOOTER' COPYRIGHT 1984 GEORG
 E JONES * FOR VIC 20                       CD
```

FASTER 64

FROM PAGE 39

First byte: C000 Last byte: C242

```
C000: 78 A9 00 85 FC A2 A0 86 6E
C008: FD A0 00 A2 00 9D 43 C2 EC
C010: 9D 43 C3 9D 43 C4 9D 43 3B
C018: C5 9D 43 C6 E8 D0 EE B1 DF
C020: FC 91 FC E6 FC D0 02 E6 49
C028: FD A5 FD C9 C0 D0 F0 A5 BB
C030: 01 29 FE 85 01 58 A2 4C 27
C038: 8E 8B B0 A2 79 8E 8C B0 EA
C040: A2 C0 8E 8D B0 A2 CA 8E 6C
C048: 55 A8 A2 C0 8E 56 A8 A2 D9
C050: F0 8E E5 B0 A2 C1 8E E6 40
C058: B0 A2 4C 8E E3 B1 A2 F8 B7
C060: 8E E4 B1 A2 C1 8E E5 B1 10
C068: A0 00 B9 0D C2 C9 00 D0 2D
C070: 01 60 20 D2 FF C8 4C 6A 44
C078: C0 A2 00 8E 07 C2 20 79 CD
C080: 00 20 90 B0 48 98 48 A2 AD
C088: 00 BD 43 C2 F0 25 BD 43 63
C090: C2 C5 45 D0 1A BD 43 C3 0E
C098: C5 46 D0 13 BD 43 C6 CD 1E
C0A0: 07 C2 D0 0B FE 43 C4 D0 1E
C0A8: 1D FE 43 C5 4C C6 C0 E8 8A
C0B0: 4C 89 C0 A5 45 9D 43 C2 D5
C0B8: A5 46 9D 43 C3 FE 43 C4 50
C0C0: AD 07 C2 9D 43 C6 68 A8 F0
C0C8: 68 60 A9 0D 20 D2 FF A2 DD
C0D0: 00 8E 07 C2 A0 00 B9 44 C7
C0D8: C2 F0 7A B9 44 C5 D9 43 E7
C0E0: C5 F0 05 B0 0D 4C 51 C1 B9
C0E8: B9 44 C4 D9 43 C4 90 61 7F
C0F0: F0 5F A9 01 8D 07 C2 B9 FC
C0F8: 43 C2 8D 08 C2 B9 43 C3 18
C100: 8D 09 C2 B9 43 C4 8D 0A B2
C108: C2 B9 43 C5 8D 0B C2 B9 A2
C110: 43 C6 8D 0C C2 B9 44 C2 37
C118: 99 43 C2 B9 44 C3 99 43 56
C120: C3 B9 44 C4 99 43 C4 B9 02
C128: 44 C5 99 43 C5 B9 44 C6 99
C130: 99 43 C6 AD 08 C2 99 44 2A
C138: C2 AD 09 C2 99 44 C3 AD C3
C140: 0A C2 99 44 C4 AD 0B C2 2B
C148: 99 44 C5 AD 0C C2 99 44 46
C150: C6 C8 4C D6 C0 AE 07 C2 3C
```

```
C158: F0 03 4C CF C0 A0 00 8C 56
C160: 06 C2 38 20 F0 FF 98 AC B7
C168: 06 C2 C9 1E 90 05 A9 0D 65
C170: 20 D2 FF B9 43 C2 F0 5A 6E
C178: 29 7F 20 D2 FF B9 43 C3 D4
C180: 29 7F 20 D2 FF B9 43 C3 DC
C188: 29 80 F0 14 B9 43 C2 29 20
C190: 80 F0 08 A9 25 20 D2 FF CB
C198: 4C A0 C1 A9 24 20 D2 FF 08
C1A0: B9 43 C6 F0 0A A9 28 20 51
C1A8: D2 FF A9 29 20 D2 FF A9 EA
C1B0: 20 20 D2 FF B9 43 C4 AA 30
C1B8: B9 43 C5 8C 06 C2 20 CD BE
C1C0: BD AC 06 C2 A9 20 20 D2 B0
C1C8: FF A9 2C 20 D2 FF C8 4C A6
C1D0: 5F C1 A9 14 20 D2 FF A9 4C
C1D8: 00 A2 00 9D 43 C2 9D 43 FF
C1E0: C3 9D 43 C4 9D 43 C5 9D 8E
C1E8: 43 C6 E8 D0 EE 4C 86 E3 52
C1F0: A9 01 8D 07 C2 4C D1 B1 C2
C1F8: AD 07 C2 48 20 B2 B1 68 A5
C200: 8D 07 C2 4C E6 B1 00 00 3C
C208: 00 00 00 00 00 0D 46 41 9C
C210: 53 54 45 52 20 36 34 20 F9
C218: 57 4F 52 4B 49 4E 47 2E 69
C220: 0D 42 59 20 41 4E 54 48 15
C228: 4F 4E 59 20 57 4F 4F 44 79
C230: 0D 48 4F 55 53 54 4F 4E 6F
C238: 2C 20 54 58 20 31 39 38 F3
C240: 34 0D 00 81
```

BASIC Trace

FROM PAGE 57

BASIC LOADER

```
•15 IFPEEK(49152)=169THENSYS49152:END            FJ
•20 PRINT"[CLEAR][WHITE][RVSON] BASIC TRA
 CE [RVSOFF]":PRINT"[DOWN]TO BEGIN TRACIN
 G YOUR BASIC PROGRAM,"                          LO
•25 PRINT"TYPE:[3" "]TRACE <RETURN>"             KO
•30 PRINT"THE COMPUTER WILL PRINT [RVSON]
 ON [RVSOFF], AND THE"                           CE
•35 PRINT"TRACE WINDOW WILL APPEAR.  NOW
 RUN YOUR"                                       IG
•40 PRINT"BASIC PROGRAM.  TO SPEED UP THE
 TRACING"                                        JM
•45 PRINT"HOLD DOWN THE [RVSON] SHIFT [RV
 SOFF] KEY."                                     MI
•50 PRINT"WHEN YOU NO LONGER WANT TRACE,
 TYPE:"                                          CI
•55 PRINT"TRACE <RETURN>. THE COMPUTER WI
 LL PRINT"                                       MC
•60 PRINT"[RVSON] OFF [RVSOFF], MEANING T
 RACE IS NOW OFF."                               IF
•65 PRINT"TO RESTART THE TRACE, TYPE:            AD
•70 PRINT"TRACE <RETURN> AGAIN."                 CO
•101 PRINT"[3"[DOWN]"]ARE YOU USING TAPE
 OR DISK[3" "](T/D)?"                            GL
•105 GETK$:ON -(K$="")GOTO105:D=1:IFK$="D
 "THEND=8                                        JC
•110 PRINT"[DOWN][DOWN]PLEASE WAIT WHILE
 I LOAD THE ML.":LOAD"BTML",D,1                  EM
```

ML LISTING

First byte: C000   Last byte: C1F8   SYS to start: RUN

```
C000: A9 4C 85 7C A9 80 85 7D 25
C008: A9 C1 85 7E 60 60 60 60 F8
C010: 8E FE C1 8C FC C1 A9 00 54
C018: 2A 8D FB C1 60 00 00 00 ED
C020: A2 23 A0 06 8C 42 C1 A0 BD
C028: 05 A9 01 9D 00 D8 E8 88 BF
C030: D0 F9 8A 18 69 23 AA CE A3
C038: 42 C1 D0 EB F0 07 FD 43 32
C040: C1 CA 18 90 43 A2 02 A5 03
C048: 39 8D 3E C1 CD 3F C1 D0 AE
C050: 01 CA A4 3A C0 FF D0 02 8E
C058: A0 00 8C 40 C1 CC 41 C1 57
C060: D0 03 CA F0 78 8D 3F C1 F6
C068: 8C 41 C1 A2 00 A0 00 8C C7
C070: 42 C1 AD 3E C1 38 FD 43 9B
C078: C1 A8 E8 AD 40 C1 30 BE 6A
C080: FD 43 C1 CA 0A B0 0C 6A 7F
C088: 8D 40 C1 8C 3E C1 EE 42 D5
C090: C1 90 DF 8A 4A AA AD 42 32
C098: C1 9D 4D C1 8A 0A AA E8 2F
C0A0: E8 E0 0A D0 C8 A9 00 8D 45
C0A8: 3D C1 AD 8D 02 C9 01 F0 A0
C0B0: 0F A9 05 8D 3D C1 A2 00 9D
C0B8: A0 00 C8 D0 FD E8 D0 F8 A3
C0C0: A2 00 BD 67 C1 9D 62 C1 0C
C0C8: E8 E0 14 D0 F5 A2 00 BD CD
C0D0: 4D C1 18 69 B0 9D 76 C1 E7
C0D8: E8 E0 05 D0 F2 A9 00 8D A2
C0E0: 5C C1 AE 3D C1 8E 5D C1 5A
C0E8: AE 5D C1 BD 52 C1 AE 5C 93
C0F0: C1 9D 23 04 EE 5C C1 EE 73
C0F8: 5D C1 AD 5C C1 C9 05 D0 83
C100: E7 A9 05 8D 3E C1 A2 23 E9
C108: 8E 40 C1 A2 00 8E 42 C1 CD
C110: A0 05 AE 42 C1 BD 62 C1 4A
C118: AE 40 C1 9D 28 04 EE 40 C1
C120: C1 EE 42 C1 88 D0 EB AD C7
C128: 40 C1 18 69 23 8D 40 C1 5E
C130: CE 3E C1 D0 DB 60 60 60 CC
C138: 00 00 00 20 C0 05 00 F5 14
C140: EB 00 19 10 27 E8 03 64 CC
C148: 00 0A 00 01 00 00 00 02 55
C150: 04 05 BC 92 95 8E BE 90 1C
C158: 81 95 93 85 05 0A 00 00 97
C160: 00 00 B1 B0 B0 B1 B8 B1 8F
C168: B0 B0 B1 B9 B1 B0 B0 B2 FA
C170: B0 B1 B0 B0 B2 B1 B0 B0 F9
C178: B2 B4 B5 00 00 00 00 00 95
C180: 8D FF C1 20 10 C0 AD FD 6C
C188: C1 F0 03 20 20 C0 A2 00 E1
C190: BD 00 02 F0 08 DD EA C1 D3
C198: D0 07 E8 D0 F3 E0 00 D0 CF
C1A0: 0E 20 F0 C1 AD FF C1 C9 BA
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 93 and 94 explain these codes and provide other essential information on entering *Ahoy!* programs. Refer to these pages **before** entering any programs!

```
C1A8: 3A B0 03 4C 80 00 60 AD 71
C1B0: FD C1 C9 01 F0 16 A9 01 EC
C1B8: 8D FD C1 A9 4F 20 D2 FF F1
C1C0: A9 4E 20 D2 FF A9 0D 20 82
C1C8: D2 FF D0 17 A9 00 8D FD B8
C1D0: C1 A9 4F 20 D2 FF A9 46 6E
C1D8: 20 D2 FF 20 D2 FF A9 0D 75
C1E0: 20 D2 FF A9 00 8D 00 02 0D
C1E8: F0 B7 54 52 41 43 45 00 02
C1F0: AE FE C1 AC FC C1 AD FB 75
C1F8: C1 6A 60 85
```

ELECHECK

FROM PAGE 43

```
•0 PRINT"[CLEAR]":POKE53281,13:POKE53280,
 7:DIMDN(50):DIMCO(50)                     DH
•1 DIMPE$(200):DIMCN(200):DIMCD$(200):DIM
 CP$(200):DIMCA(200):DIMM$(14)             ND
•2 PRINT"[BLACK]":PRINT:PRINT"[8" "]PLEAS
 E ENTER CURRENT DATE":PRINT               JG
•3 PRINT"[14" "][RVSON]<MM-DD-YY>[RVSOFF]
 ";DA$                                     EB
•4 PRINT:INPUT"[13" "]";DA$:IFLEN(DA$)<>8
 THENPRINT"WATCH FORMAT":GOTO3             NF
•6 BB$="[40"*"]"                           NG
•7 VV$="[40"="]"                           IG
•8 FF$="[40"="]"                           GG
•9 YY$=VV$+FF$                             NH
•10 POKE53281,7:POKE53280,7:PRINT"[CLEAR]
 ":FORP=1TO3:PRINT:NEXT                    DN
•11 POKE53282,16:POKE53283,15:POKE53284,2
 :POKE53265,PEEK(53265)OR64                AK
•20 PRINTTAB(8);"[RVSON][BLACK]1. ADD CHE
 CK INFORMATION":PRINT                     KC
•30 PRINTTAB(8);"[RVSON][BLACK]2. EXAMINE
 CHECK FILE[3" "]":PRINT                   CM
•40 PRINTTAB(8);"[RVSON][BLACK]3. MENU OF
 PRINT OPTIONS":PRINT                      NB
•50 PRINTTAB(8);"[RVSON][BLACK]4. CHANGE
 INFORMATION[3" "]":PRINT                  HK
•60 PRINTTAB(8);"[RVSON][BLACK]5. INITIAL
 IZE OPTIONS[3" "]":PRINT                  NL
•65 PRINTTAB(8);"[RVSON][BLACK]6. BALANCE
 CHECK BOOK[3" "]":PRINT                   CE
•70 PRINTTAB(8);"[RVSON][BLACK]7. GRAPH C
 OMPARSIONS[4" "]":PRINT                   HI
•75 PRINTTAB(8);"[RVSON][BLACK]8. END PRO
 GRAM[10" "]":PRINT                        GG
•80 PRINTTAB(8);"[RVSON][BLACK]PLEASE SEL
 ECT ONE[7" "]"                            EK
•85 GETX$:IFX$=""THEN85                    FI
•86 IFX$<"1"ORX$>"8"THENPRINT"ENTER A NUM
 BER 1-7":GOTO85                           BD
•90 X=VAL(X$):ONXGOTO30100,30100,31300,30
 100,20000,6000,30100,32000                IK
•100 S$="":K=0:C=0:H=0:Q=0:T=0:X=0:A=0:GT
 =0:PRINT"[CLEAR]":GOTO10                  JP
•105 REM*********************************
 *                                         EO
•110 REM**WRITTEN BY GLENN LUMPKINS        BK
•140 REM*********************************
 *                                         EO
•6000 REM*** ROUTINE TO BALANCE CHECK BOO
 K***                                      GE
•6001 POKE53281,7:POKE53280,9              DH
•6010 :PRINT"[CLEAR]":PRINT"[6" "]ENTER B
 ANK'S ENDING BALANCE"                     DN
•6015 PRINT:INPUT"[4" "]$ ";BE             DD
•6020 PRINT:PRINT"[6" "]HOW MANY DEPOSITS
 NOT LISTED":PRINT                         JH
•6025 INPUT"[4" "]$ ";D$:PRINT"[CLEAR]":D
 =VAL(D$)                                  JL
•6030 IFD$=""ORD$="0"THEN6080              KK
•6040 FORI=1TOD                            JP
•6050 PRINT"[7" "]ENTER DEPOSIT NOT LISTE
 D":PRINT:INPUT"[5" "]$ ";DN(I)            IK
•6055 PRINT"[CLEAR]"                       HH
•6060 DP=DP+DN(I):NEXT                     IG
•6080 PRINT"[6" "]HOW MANY CHECKS OUTSTAN
 DING"                                     DB
•6085 PRINT:INPUT"[3" "]$ ";CN$:PRINT"[CL
 EAR]":CN=VAL(CN$)                         BA
•6090 IFCN$="0"ORCN$=""THEN6130            FD
•6100 FORI=1TOCN                           OF
•6110 PRINT"[3" "]ENTER AMOUNT OF OUTSTAN
 DING CHECK"                               DM
•6115 PRINT:INPUT"$ ";CO(I):PRINT"[CLEAR]
 "                                         EA
•6120 AC=AC+CO(I):NEXT                     FK
•6130 PRINT"[7" "]ANY OTHER CHARGES SUCH
 AS:"                                      HF
•6135 PRINT:PRINT"[10" "]AUTOMATIC DEDUCT
 IONS":PRINT                               MN
•6137 PRINT"[12" "]SERVICE CHARGES"        HM
•6140 PRINT:PRINT"[17" "][RVSON]<Y/N>[RVS
 OFF]"                                     KM
•6142 GETAS$:IFAS$=""THEN6142              EA
•6145 IFAS$="N"THEN6230                    ON
•6150 IFAS$="Y"THEN6200                    BP
•6155 IFAS$<>"N"ORAS$<>"Y"THEN6140         NB
•6200 PRINT"[CLEAR]":PRINT"[3" "]ENTER IN
 TOTAL AMOUNT OF CHARGES"                  CB
•6220 PRINT:INPUT"  $ ";OC                 AL
•6230 BA=(BE+DP)-(AC+OC)                   OP
•6240 PRINT"[CLEAR]":PRINT:PRINT"[6" "]EN
 TER YOUR CHECKBOOK BALANCE"               FM
•6250 PRINT:INPUT"[5" "]$ ";YB:PRINT"[CLE
 AR]"                                      PD
•6255 IFYB=BATHEN6300                      NM
•6260 IFBA>0THENPRINT"[10" "]YOUR BALANCE
 SHOULD BE":PRINT                          OF
```

```
·6262 BN$=STR$(BA+.001):BN$=LEFT$(BN$,LEN
 (BN$)-1)                                  BE
·6265 IFBA<0THENPRINT"[10" "]YOUR BALANCE
 SHOULD BE":PRINT                          GP
·6266 IFBA>0THENPRINT"[9" "]$";BN$:PRINT:
 GOTO6280                                  OO
·6270 B$=STR$(BA-.001):B$=LEFT$(B$,LEN(B$
 )-1):PRINT"[10" "]$";B$:PRINT             JG
·6280 IFBA<YBTHENBX=YB-BA:PRINT:PRINT"[10
 " "]YOU ARE OVER BY"                      MJ
·6285 BC$=STR$(BX+.001):BC$=LEFT$(BC$,LEN
 (BC$)-1)                                  CH
·6288 IFBA<YBTHEN:PRINT:PRINT"[9" "][SS]$
 ";BC$:GOTO7000                            KK
·6290 IFBA>YBTHENAX=BA-YB:PRINT:PRINT"[10
 " "]YOU ARE UNDER BY"                     MP
·6295 AY$=STR$(AX+.001):AY$=LEFT$(AY$,LEN
 (AY$)-1)                                  AO
·6297 IFBA>YBTHENPRINT:PRINT"[10" "]$";AY
 $                                         BD
·6300 IFBA=YBTHENPRINT:PRINT"[CLEAR]":PRI
 NT"YOU ARE IN BALANCE--NO ADJUSTMENT NEE
 DED"                                      AK
·7000 PRINT:PRINT:PRINT:PRINT"[7" "]PRESS
 ANY KEY TO CONTINUE"                      FB
·7010 GETP$:IFP$=""THEN7010                LK
·7012 BA=0:YB=0:AC=0:DP=0                  MF
·7020 GOTO100                              CF
·10000 REM***ROUTINE FOR CHANGE***         GL
·10001 POKE53281,1:POKE53280,8             BO
·10010 PRINT"[CLEAR]":OPEN5,8,5,"0:"+Z$+"
 ,R"                                       NA
·10015 PRINT"[4" "]ONE MOMENT READING ";Z
 $                                         HJ
·10020 FORI=1TO199:C=I                     GJ
·10030 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
 ,CA(I)                                    CK
·10040 IFCN(I)=9999THENCLOSE5:GOTO10060    OK
·10050 NEXT                                IA
·10060 PRINT"[CLEAR]":PRINTTAB(11);Z$:PRI
 NT                                        OA
·10065 PRINTBB$                            EA
·10070 PRINT"[RVSON]R#";"[RVSOFF]  ";"[RV
 SON]CK NO.";"[RVSOFF][3" "]";"[RVSON]DAT
 E";"[RVSOFF][4" "]";"[RVSON]PAYEE":PRINT  OF
·10071 IFK=1THEN10100                      FK
·10090 FORI=1TOC:K=K+1                     OK
·10100 PRINTI;TAB(4)CN(I);TAB(11)CD$(I);T
 AB(21)PE$(I)                              OH
·10101 V$=STR$(CA(I)+.001):V$=LEFT$(V$,LE
 N(V$)-1)                                  DJ
·10102 PRINT"[RVSON]PURPOSE[s Z][RVSOFF]"
 ;TAB(11)CP$(I)                            JN
·10106 PRINT"[RVSON]AMOUNT [s Z][RVSOFF]"
 ;TAB(11)"$";V$:PRINT                      JI
·10120 IFK=4THENPRINT:PRINT"[9" "]CONTINU
 E PAGING [RVSON]<Y/N>[RVSOFF]"            BO
·10121 IFK=4THENGETSS$:IFSS$=""THEN10121   JA
·10125 IFK>4THENK=1:GOTO10060              HP
·10135 IFSS$="N"THENGOTO10150              KN
·10140 NEXT                                IA
·10150 PRINT:INPUT"CHANGE WHICH RECORD NO
 .";RN                                     PC
·10160 PRINT"[CLEAR]":PRINT"RECORD NUMBER
 ";RN:PRINT                                AC
·10170 PRINT"1. CHECK NUMBER ";CN(RN):PRI
 NT                                        GL
·10171 PRINT"2. CHECK DATE[SS][SS]  ";CD$
 (RN):PRINT                                AL
·10172 PRINT"3. CHECK PAYEE[3" "]";PE$(RN
 ):PRINT                                   LD
·10174 PRINT"4. CHECK PURPOSE ";CP$(RN):P
 RINT                                      PE
·10176 PRINT"5. CHECK AMOUNT ";CA(RN):PRI
 NT                                        EB
·10177 PRINT"6. RETURN TO MENU ":PRINT     DB
·10200 PRINT:PRINT"CHANGE WHICH LINE"      CJ
·10201 GETCI$:IFCI$=""THEN10201            KD
·10202 CI=VAL(CI$):IFCI<1ORCI>6THENPRINT"
 ENTER A NUMBER 1-6":GOTO10201             LI
·10205 PRINT"[CLEAR]":FORG=1TO10:PRINT:NE
 XTG:PRINTTAB(7);Z$;" CHECK"               HP
·10210 IFCI=1THENPRINT"OLD CHECK NUMBER "
 ;CN(RN):INPUT"NEW CHECK NUMBER";CN(RN)    PM
·10219 IFCI=6THENPRINT"[CLEAR]":GOTO100    JF
·10220 IFCI=2THENPRINT"OLD CHECK DATE  ";
 CD$(RN):INPUT"NEW CHECK DATE";CD$(RN)     NE
·10225 IFLEN(CD$(RN))>8THENPRINT"WATCH FO
 RMAT":PRINT:GOTO10220                     EM
·10230 IFCI=3THENPRINT"OLD CHECK PAYEE  "
 ;PE$(RN):INPUT"NEW CHECK PAYEE";PE$(RN)   KI
·10235 IFLEN(PE$(RN))>17THENPRINT"17 CHAR
 ACTERS MAX. PLEASE":PRINT:GOTO10230       AC
·10240 IFCI=4THENPRINT"OLD CHECK PURPOSE
 ";CP$(RN)                                 NK
·10241 IFCI=4THENINPUT"NEW CHECK PURPOSE
 ";CP$(RN)                                 HE
·10245 IFLEN(CP$(RN))>30THENPRINT"30 CHAR
 ACTERS MAX. PLEASE":PRINT:GOTO10240       HH
·10250 IFCI=5THENPRINT"OLD CHECK AMOUNT "
 CA(RN):INPUT"NEW CHECK AMOUNT";CA(RN)     GH
·10252 PRINT:PRINT"CHANGE ANOTHER LINE [R
 VSON]<Y/N>[RVSOFF]"                       PK
·10253 GETAC$:IFAC$=""THEN10253            CC
·10255 IFAC$="N"THENGOTO10260              GL
·10256 IFAC$="Y"THENGOTO10160              KH
·10260 IFX=4THENOPEN5,8,5,"@0:"+Z$+",W"    IA
·10270 FORI=1TOC                           JM
·10280 PRINT#5,CN(I):PRINT#5,CD$(I):PRINT
 #5,PE$(I):PRINT#5,CP$(I):PRINT#5,CA(I)    PG
·10290 NEXT:I=C+1:CN(I)=999:CD$(I)="DATE"
 :PE$(I)="EOF":CP$(I)="[4"X"]":CA(I)=9999  NC
·10300 PRINT#5,CN(I):PRINT#5,CD$(I):PRINT
 #5,PE$(I):PRINT#5,CP$(I):PRINT#5,CA(I)    PG
·10310 CLOSE5:GOTO100                      CK
·19999 REM***ROUTINE FOR INITIALIZING**    OO
```

```
•20000 PRINT"[CLEAR]":FORG=1TO8:PRINT:NEX
T                                         MB
•20001 PRINTTAB(6);"[RVSON]1. INITIALIZE
WITH INFORMATION":PRINT                   OD
•20002 PRINTTAB(6);"[RVSON]2. INITIALIZE
ONE FILE ONLY[3" "]":PRINT                GM
•20003 PRINTTAB(6);"[RVSON]3. AUTO INITIA
LIZE ALL FILES  ":PRINT                   MH
•20004 PRINTTAB(6);"[RVSON]4. RETURN TO M
ENU[13" "]":PRINT                         DJ
•20005 PRINTTAB(6);"[RVSON]PLEASE SELECT
ONE[13" "]"                               KO
•20006 GETQ$:IFQ$=""THEN20006             AG
•20007 Q=VAL(Q$):IFQ<1ORQ>4THEN20006      ON
•20010 IFQ=1THENH=0:C=0:GOTO30100         JF
•20025 IFQ=4THENGOTO100                   IB
•20030 IFQ=2THENGOTO30100                 PJ
•20040 PRINT"[CLEAR]":FORG=1TO12:PRINT:NE
XT:PRINTTAB(7);"INITIALIZING ";Z$         IO
•20042 OPEN5,8,5,"0:"+Z$+",W"             EB
•20050 PE$(1)="EOF":CN(1)=9999:CD$(1)="DA
TE":CP$(1)="[4"X"]":CA(1)=9999            JI
•20060 PRINT#5,CN(1):PRINT#5,CD$(1):PRINT
#5,PE$(1):PRINT#5,CP$(1):PRINT#5,CA(1)    OO
•20070 CLOSE5:IFQ=2THENQ=0:A=0:GOTO100    DP
•20080 IFQ=3THENA=A+1:GOTO30115           GK
•29999 REM***DATA FOR SUB MENU***         DP
•30000 DATASELECTION MENU,1.  CHARGE CARD
S,2.  CONTRIBUTIONS                       IG
•30010 DATA3.  ENTERTAINMENT,4.  HOUSEHOL
D,5.  LOAN PAYMENTS,6.  MEDICAL/DENTAL    AG
•30020 DATA7.  MORTGAGE PAYMENTS,8.  TAXE
S,9.  TUITION,10. UTILITIES               LC
•30030 DATA11. INSURANCE,12. MISCELLANEOU
S,13. RETURN MAIN MENU                    GJ
•30100 PRINT"[CLEAR]":FORS=1TO4:PRINT:NEX
T                                         PB
•30105 REM**ROUTINE TO SELECT  ROUTINES** FO
•30110 FORS=1TO14:READM$(S):PRINTTAB(9);M
$(S):NEXT:RESTORE                         EC
•30112 PRINT:INPUT"[7" "]PLEASE SELECT ON
E";A                                      GC
•30113 IFA<1ORA>13THEN30112               HJ
•30115 IFA=1THENZ$="CHARGE CARDS"         KO
•30117 IFA=2THENZ$="CONTRIBUTIONS"        HN
•30119 IFA=3THENZ$="ENTERTAINMENT"        PH
•30120 IFA=4THENZ$="HOUSEHOLD"            HJ
•30122 IFA=5THENZ$="LOAN PAYMENTS"        AH
•30124 IFA=6THENZ$="MEDICAL/DENTAL"       LC
•30126 IFA=7THENZ$="MORTGAGE PAYMENTS"    FD
•30128 IFA=8THENZ$="TAXES"                EH
•30130 IFA=9THENZ$="TUITION"              OE
•30132 IFA=10THENZ$="UTILITIES"           KL
•30134 IFA=11THENZ$="INSURANCE"           LG
•30136 IFA=12THENZ$="MISCELLANEOUS"       KN
•30137 IFA=13THENOPEN3,4:GOTO31393        ID
•30138 IFQ=2THENGOTO20040                 PF
•30139 IFQ=3THENGOTO20040                 AC
•30140 IFX=1THENGOTO31000                 PJ
•30145 IFX=2THENGOTO31200                 AA
•30150 IFX=4THENGOTO10000                 OF
•30155 IFX=3THENGOTO31305                 PN
•30158 IFX=7THENGOTO35000                 PL
•30160 IFH=2THENGOTO31335                 PN
•30170 IFH=1THEN31450                     GK
•30180 GOTO31070                          IE
•31000 REM**ROUTINE TO READ--LOAD ARRAY--
ADD INFO***                               JP
•31001 POKE53281,12:POKE53280,10:PRINT"[C
LEAR]":PRINT"[4" "]ONE MOMENT READING "Z
$                                         GB
•31010 OPEN5,8,5,"0:"+Z$+",R"             GG
•31020 FORI=1TO199:C=I                    GJ
•31030 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
,CA(I)                                    CK
•31040 IFCN(I)=9999THENCLOSE5:GOTO31055   NJ
•31050 NEXT                               IA
•31055 C=C-1                              BM
•31070 PRINT"[CLEAR]":C=C+1:IFX=5THENPRIN
T"INITIALIZING ";Z$                       GB
•31071 PRINT:PRINT:IFX=1THEN PRINT"INFORM
ATION FOR ";Z$:PRINT                      DH
•31075 INPUT"ENTER CHECK NUMBER[3" "]";CN
(C):PRINT                                 BP
•31080 INPUT"ENTER DATE <MM-DD-YY>";CD$(C
):PRINT                                   KO
•31085 IFLEN(CD$(C))<>8THENPRINT"WATCH FO
RMAT":GOTO31080                           GJ
•31090 INPUT"ENTER PAYEE[10" "]";PE$(C):P
RINT                                      JB
•31095 IFLEN(PE$(C))>17THENPRINT"17 CHARA
CTERS MAX. PLEASE":GOTO31090              AE
•31100 INPUT"ENTER PURPOSE[8" "]";CP$(C):
PRINT                                     CN
•31105 IFLEN(CP$(C))>30THENPRINT"30 CHARA
CTERS MAX. PLEASE":GOTO31100              DI
•31110 INPUT"ENTER CHECK AMOUNT[3" "]";CA
(C):PRINT                                 IN
•31115 PRINT"IS THIS CORRECT [RVSON]<Y/N/
R>[RVSOFF]":PRINT                         ND
•31116 GETR$:IFR$=""THEN31116             OA
•31117 IFR$="N"THEN31075                  OM
•31118 IFR$="R"THENGOTO100                AB
•31120 PRINT"ANOTHER ";Z$;" CHECK [RVSON]
<Y/N>[RVSOFF] "                           OE
•31130 GETAN$:IFAN$=""THEN31130           JH
•31140 IFAN$="Y"THEN31070                 FP
•31142 IFX=5THENOPEN5,8,5,"0:"+Z$+",W":GO
TO31150                                   PP
•31145 OPEN5,8,5,"@0:"+Z$+",W"            CL
•31150 FORI=1TOC                          JM
•31160 PRINT#5,CN(I):PRINT#5,CD$(I):PRINT
#5,PE$(I):PRINT#5,CP$(I):PRINT#5,CA(I)    PG
•31170 NEXT:I=C+1:PE$(I)="EOF":CN(I)=9999
:CD$(I)="DATE":CP$(I)="[4"X"]":CA(I)=999
9                                         OL
```

```
•31180 PRINT#5,CN(I):PRINT#5,CD$(I):PRINT
 #5,PE$(I):PRINT#5,CP$(I):PRINT#5,CA(I)    PG
•31182 CLOSE5:GOTO100                      CK
•31200 REM***ROUTINE TO EXAMINE FILES***   AL
•31201 POKE53281,7:POKE53280,14            HD
•31210 OPEN5,8,5,"0:"+Z$+",R":PRINT"[CLEA
 R]":PRINT"[4" "]ONE MOMENT READING ";Z$   LM
•31215 FORI=1TO199:C=I                     GJ
•31220 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
 ,CA(I)                                    CK
•31225 IFCN(I)=9999THENCLOSE5:GOTO31232    OI
•31230 NEXT                                IA
•31232 PRINT"[CLEAR]":PRINTTAB(11);Z$:PRI
 NT                                        OA
•31233 PRINTBB$                            EA
•31235 PRINT"[RVSON]CK NO.";"[RVSOFF][5"
 "]";"[RVSON] DATE ";"[RVSOFF][5" "]";"[R
 VSON] PAYEE ":PRINT                       NE
•31237 IFK=1THENGOTO31242                  OC
•31240 FORI=1TOC-1:K=K+1                   OP
•31242 V$=STR$(CA(I)+.001):V$=LEFT$(V$,LE
 N(V$)-1)                                  DJ
•31245 PRINTCN(I),CD$(I),"  ";PE$(I)       JD
•31246 PRINT"[RVSON]PURPOSE[s Z][RVSOFF]"
 ,CP$(I)                                   FM
•31247 PRINT"[RVSON]AMOUNT [s Z][RVSOFF]"
 ,"$";V$:PRINT                             HB
•31250 IFK=4THENPRINT:PRINT"[9" "]CONTINU
 E PAGING [RVSON]<Y/N>[RVSOFF]"            BO
•31251 IFK=4THENGETS$:IFS$=""THEN31251     JK
•31253 IFK>4THENK=1:GOTO31232              JN
•31255 IFS$="N"THENGOTO100                 PG
•31260 NEXT                                IA
•31269 PRINT:PRINT"[7" "]PRESS ANY KEY TO
  CONTINUE"                                JF
•31270 GETCT$:IFCT$=""THEN31270            GA
•31271 GOTO100                             CF
•31300 PRINT"[CLEAR]":FORP=1TO8:PRINT:NEX
 TP:POKE53281,3:POKE53280,10               GN
•31305 PRINTTAB(9);"[RVSON]PRINT OPTIONS[
 5" "]":PRINT:PRINTTAB(9);"[RVSON]1. ONE
 CHECK FILE "                              LO
•31307 PRINT:PRINTTAB(9)"[RVSON]2. ALL CH
 ECK FILES"                                GE
•31308 PRINT:PRINTTAB(9);"[RVSON]3. RETUR
 N TO MENU ":PRINT                         KL
•31310 PRINTTAB(9);"[RVSON]PLEASE SELECT
 ONE "                                     PD
•31312 GETH$:IFH$=""THEN31312              MK
•31313 IFH$<"1"ORH$>"3"THENH$="":GOTO3131
 0                                         PO
•31314 IFH$="2"THENPRINT"[CLEAR]":PRINT"[
 5" "]ARE YOU SURE YOU WANT ALL FILES":PR
 INT                                       BA
•31315 IFH$="2"THENINPUT"[17" "][RVSON]<Y
 /N/R>[RVSOFF]";IC$                        PL
•31316 IFIC$="N"THEN31300                  CF
•31317 IFIC$="R"THEN100                    KI
•31320 H=VAL(H$):IFH<1ORH>3THEN31315       JP
•31331 A=0:T=0:X=0:C=0:Q=0:GT=0:PRINT"[CL
 EAR]"                                     EA
•31332 IFH=2THENA=A+1:GOTO30115            DK
•31333 IFH=1THENGOTO30100                  OJ
•31334 IFH=3THENPRINT"[CLEAR]":GOTO100     DP
•31335 REM**START OF PRINT ALL FILES**     DN
•31336 OPEN5,8,5,"0:"+Z$+",R"              GG
•31338 IFH=2THENPRINTTAB(7);"PRINTING ";Z
 $;" FILE":PRINT                           CD
•31340 FORI=1TO199:C=I                     GJ
•31345 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
 ,CA(I)                                    CK
•31350 IFCN(I)=9999THENCLOSE5:GOTO31368    NI
•31360 T=T+CA(I)                           CG
•31365 NEXT                                IA
•31366 REM**ROUTINE TO PRINT ALL FILES**   LC
•31368 C=C-1:GT=GT+T:GT$=STR$(GT+.001):GT
 $=LEFT$(GT$,LEN(GT$)-1)                   LB
•31369 IFCN(1)=9999THENPRINTTAB(7);"NO EN
 TRIES FOR ";Z$:CLOSE5:PRINT:GOTO31332     MH
•31370 OPEN3,4:PRINT#3,CHR$(16)CHR$(51)CH
 R$(51);Z$                                 NL
•31373 CC=LEN(Z$):UU$=LEFT$(BB$,CC):PRINT
 #3,CHR$(16)CHR$(51)CHR$(51);UU$           CM
•31374 PRINT#3,CHR$(10):PRINT#3,YY$:PRINT
 #3,CHR$(16)CHR$(48)CHR$(49)"CHECK #";     GA
•31375 PRINT#3,CHR$(16)CHR$(49)CHR$(50)"D
 ATE";CHR$(16)CHR$(50)CHR$(49)"[s G]";     OD
•31376 PRINT#3,CHR$(16)CHR$(50)CHR$(56)"P
 AYEE";CHR$(16)CHR$(51)CHR$(56)"[s H]";    MA
•31377 PRINT#3,CHR$(16)CHR$(52)CHR$(49)"[
 s G]";CHR$(16)CHR$(53)CHR$(50)"PURPOSE";  DG
•31378 PRINT#3,CHR$(16)CHR$(55)CHR$(48)"[
 s H]";CHR$(16)CHR$(55)CHR$(52)"AMOUNT"    HJ
•31379 PRINT#3,YY$                         NL
•31380 FORI=1TOC                           JM
•31382 PRINT#3,CHR$(16)CHR$(48)CHR$(50);C
 N(I);CHR$(16)CHR$(49)CHR$(48);CD$(I);     EK
•31384 PRINT#3,CHR$(16)CHR$(50)CHR$(49);P
 E$(I);CHR$(16)CHR$(52)CHR$(49);CP$(I);    PB
•31385 I$=STR$(INT(CA(I)))+"."+RIGHT$(STR
 $(CA(I)*100),2)                           BB
•31386 PRINT#3,CHR$(16)CHR$(55)CHR$(49)"$
 ";""TAB(8-LEN(I$));I$                     DF
•31387 T$=STR$(T+.001):T$=LEFT$(T$,LEN(T$
 )-1)                                      GN
•31392 IFI=CTHENPRINT#3:PRINT#3,"TOTAL FO
 R ";Z$;" $";T$:T=0:PRINT#3:PRINT#3        EG
•31393 IFA=13THENPRINT#3,"YEAR TO DATE ";
 DA$;" TOTAL";" $";GT$:PRINT#3:PRINT#3     DC
•31394 IFA=13THENCLOSE3:GOTO100            KG
•31395 IFI=CTHENCLOSE3:GOTO31332           OK
•31396 NEXT                                IA
•31450 REM**ROUTINE TO PRINT ONE FILE**    OF
•31460 OPEN5,8,5,"0:"+Z$+",R"              GG
•31461 IFH=1THENPRINT"[CLEAR]":FORG=1TO10
 :PRINT:NEXT:PRINTTAB(7);"PRINTING ";Z$;"
```

```
 FILE"                                      PK
·31470 FORI=1TO199:C=I                       GJ
·31480 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
 ,CA(I)                                      CK
·31490 IFCN(I)=9999THENCLOSE5:GOTO31510      OP
·31495 T=T+CA(I)                             CG
·31500 NEXT                                  IA
·31510 C=C-1:T$=STR$(T +.001):T$=LEFT$(T$
 ,LEN(T$)-1)                                 LO
·31520 OPEN3,4:PRINT#3,CHR$(16)CHR$(51)CH
 R$(51);Z$                                   NL
·31522 CC=LEN(Z$):UU$=LEFT$(BB$,CC):PRINT
 #3,CHR$(16)CHR$(51)CHR$(51);UU$             CM
·31523 PRINT#3,CHR$(10):PRINT#3,YY$:PRINT
 #3,CHR$(16)CHR$(48)CHR$(49)"CHECK #";       GA
·31524 PRINT#3,CHR$(16)CHR$(49)CHR$(50)"D
 ATE";CHR$(16)CHR$(50)CHR$(49)"[s G]";       OD
·31525 PRINT#3,CHR$(16)CHR$(50)CHR$(56)"P
 AYEE";CHR$(16)CHR$(51)CHR$(56)"[s H]";      MA
·31526 PRINT#3,CHR$(16)CHR$(52)CHR$(49)"[
 s G]";CHR$(16)CHR$(53)CHR$(50)"PURPOSE";    DG
·31527 PRINT#3,CHR$(16)CHR$(55)CHR$(48)"[
 s H]";CHR$(16)CHR$(55)CHR$(52)"AMOUNT"      HJ
·31545 PRINT#3,YY$                           NL
·31550 FORI=1TOC                             JM
·31560 PRINT#3,CHR$(16)CHR$(48)CHR$(50);C
 N(I);CHR$(16)CHR$(49)CHR$(48);CD$(I);       EK
·31561 PRINT#3,CHR$(16)CHR$(50)CHR$(49);P
 E$(I);CHR$(16)CHR$(52)CHR$(49);CP$(I);      PB
·31562 I$=STR$(INT(CA(I)))+"."+RIGHT$(STR
 $(CA(I)*100),2)                             BB
·31563 PRINT#3,CHR$(16)CHR$(55)CHR$(49)"$
 ";""TAB(8-LEN(I$));I$                       DF
·31572 IFI=CTHENGOTO31590                    PC
·31580 NEXT                                  IA
·31590 PRINT#3:PRINT#3,"TOTAL FOR ";Z$;"
 $";T$:FORL=1TO5:PRINT#3:NEXT:CLOSE3         DD
·31595 GOTO100                               CF
·32000 PRINT"[CLEAR]":FORL=1TO10:PRINT:NE
 XT                                          AL
·32005 PRINTTAB(9);"IT'S BEEN A PLEASURE"
 :FORL=1TO4000:NEXT:PRINT"[CLEAR]":END       OF
·35000 REM**ROUTINE FOR GRAPHS**             MF
·35001 PRINT"[CLEAR]": PRINT" DO YOU WISH
 A COPY OF THE GRAPH [RVSON]<Y/N>[RVSOFF
 ]"                                          HH
·35002 GETHA$:IFHA$=""THEN35002              GD
·35010 PRINT"[CLEAR]":POKE53280,7:POKE532
 81,7:PRINT"[4" "]ONE MOMENT READING "Z$     IE
·35020 OPEN5,8,5,"0:"+Z$+",R"                GG
·35030 FORI=1TO199                           PL
·35040 INPUT#5,CN(I),CD$(I),PE$(I),CP$(I)
 ,CA(I)                                      CK
·35050 IFCN(I)=9999THENCLOSE5:GOTO40005      OM
·35060 J$=LEFT$(CD$(I),2):IFJ$="01"THENJ=
 J+CA(I)                                     CL
·35070 IFJ$="02"THENF=F+CA(I)                PI
·35080 IFJ$="03"THENM=M+CA(I)                AH
·35090 IFJ$="04"THENAP=AP+CA(I)              ID
·35100 IFJ$="05"THENMA=MA+CA(I)              KG
·35110 IFJ$="06"THENJU=JU+CA(I)              LL
·35120 IFJ$="07"THENJL=JL+CA(I)              JC
·35130 IFJ$="08"THENAU=AU+CA(I)              LJ
·35140 IFJ$="09"THENSP=SP+CA(I)              AA
·35150 IFJ$="10"THENOC=OC+CA(I)              KG
·35160 IFJ$="11"THENNO=NO+CA(I)              KH
·35170 IFJ$="12"THENDE=DE+CA(I)              MA
·35180 NEXT                                  IA
·40005 PRINT"[CLEAR]"                        HH
·40010 PRINT"[3" "][s B]":N=25:IFJ>660THE
 NN=50                                       HC
·40020 PRINT"JAN[c Q][s *]";:FORI=1TOJ/N:
 PRINT"[BLUE][RVSON] [BLACK][RVSOFF]";:NE
 XT:PRINTJ;:PRINT:N=25                       MK
·40025 PRINT"[3" "][s B]":IFF>660THENN=50    HD
·40030 PRINT"FEB[c Q][s *]";:FORI=1TOF/N:
 PRINT"[RED][RVSON] [BLACK][RVSOFF]";:NEX
 T:PRINTF;:PRINT:N=25                        JP
·40040 PRINT"[3" "][s B]":IFM>660THENN=50    IE
·40050 PRINT"MAR[c Q][s *]";:FORI=1TOM/N:
 PRINT"[GREEN][RVSON] [BLACK][RVSOFF]";:N
 EXT:PRINTM;:PRINT:N=25                      HK
·40060 PRINT"[3" "][s B]":IFAP>660THENN=5
 0                                           KA
·40070 PRINT"APR[c Q][s *]";:FORI=1TOAP/N
 :PRINT"[YELLOW][RVSON] [BLACK][RVSOFF]";
 :NEXT:PRINTAP;:PRINT:N=25                   BF
·40080 PRINT"[3" "][s B]":IFMA>660THENN=5
 0                                           KF
·40090 PRINT"MAY[c Q][s *]";:FORI=1TOMA/N
 :PRINT"[PURPLE][RVSON] [BLACK][RVSOFF]";
 :NEXT:PRINTMA;:PRINT:N=25                   KP
·40100 PRINT"[3" "][s B]":IFJU>660THENN=5
 0                                           AO
·40110 PRINT"JUN[c Q][s *]";:FORI=1TOJU/N
 :PRINT"[BLACK][RVSON] [BLACK][RVSOFF]";:
 NEXT:PRINTJU;:PRINT:N=25                    KD
·40120 PRINT"[3" "][s B]":IFJL>660THENN=5
 0                                           IB
·40130 PRINT"JUL[c Q][s *]";:FORI=1TOJL/N
 :PRINT"[BLUE][RVSON] [BLACK][RVSOFF]";:N
 EXT:PRINTJL;:PRINT:N=25                     GD
·40140 PRINT"[3" "][s B]":IFAU>660THENN=5
 0                                           LN
·40150 PRINT"AUG[c Q][s *]";:FORI=1TOAU/N
 :PRINT"[RED][RVSON] [BLACK][RVSOFF]";:NE
 XT:PRINTAU;:PRINT:N=25                      KK
·40160 PRINT"[3" "][s B]":IFSP>660THENN=5
 0                                           MC
·40170 PRINT"SEP[c Q][s *]";:FORI=1TOSP/N
 :PRINT"[GREEN][RVSON] [BLACK][RVSOFF]";:
 NEXT:PRINTSP;:PRINT:N=25                    AD
·40180 PRINT"[3" "][s B]":IFOC>660THENN=5
 0                                           LN
·40190 PRINT"OCT[c Q][s *]";:FORI=1TOOC/N
 :PRINT"[YELLOW][RVSON] [BLACK][RVSOFF]";
```

```
:NEXT:PRINTOC;:PRINT:N=25                          NA
•40200 PRINT"[3" "][s B]":IFNO>660THENN=5
0                                                  PA
•40210 PRINT"NOV[c Q][s *]";:FORI=1TONO/N
:PRINT"[PURPLE][RVSON] [BLACK][RVSOFF]";
:NEXT:PRINTNO;:PRINT:N=25                          JN
•40220 PRINT"[3" "][s B]":IFDE>660THENN=5
0                                                  FI
•40230 PRINT"DEC[c Q][s *]";:FORI=1TODE/N
:PRINT"[BLACK][RVSON] [BLACK][RVSOFF]";:
NEXT:PRINTDE;:PRINT:N=25                           PC
•40235 IFHA$="Y"THENPRINT"[3" "][c Z][s C
][c E][s C][c E][s C][c E][s C][c E][s C
][c E][s C]";Z$;"[c E][s C][c E][s C][c
E][s C][c E][s C][c E][s C][c X]":GOTO40
260                                                FL
•40240 PRINT"[7"."]PRESS ANY KEY TO CONTI
NUE[6"."]"                                         PB
•40250 GETA$:IFA$=""THEN40250                      NJ
•40260 J=0:F=0:M=0:AP=0:MA=0:JU=0:JL=0:AU
=0:SP=0:OC=0:NO=0:DE=0                             DC
•40270 IFHA$="Y"THEN41001                          BM
•41000 IFHA$="N"THENGOTO100                        ED
•41001 PRINT"[HOME]";:SS=(PEEK(210))*256:
OPEN3,3:OPEN4,4                                    AL
•41002 FORR=0TO24:B$=""                            DN
•41003 FORC=0TO39:A$=""                            DN
•41004 IFPEEK(SS+((R*40)+C))>127THEN:GET#
3,A$:B$=B$+CHR$(18)+A$+CHR$(146)                   CP
•41005 IFPEEK(SS+((R*40)+C))>127THENGOTO4
1100                                               JJ
•41006 GET#3,A$:IFA$=CHR$(13)THEN:A$=" "           CG
•41007 B$=B$+A$                                    IG
•41100 NEXTC:PRINT#4,B$:NEXTR:CLOSE4:CLOS
E3:GOTO100                                         HG
```

Roll Over Pachelbel

FROM PAGE 91

```
•10 REM**READ BASE FREQS**                         DN
•12 DIMF(11):FORI=0TO11:READF(I):NEXT              PO
•14 REM**READ TITLE & VARIABLES**                  CF
•16 READZ$,QM,QN,P1,P2:P3=P1-P2:IFP3<1THE
NP3=1                                              CI
•18 DIMB(QM):B=49152:FORI=0TOQM:B(I)=B+QN
*I*2:NEXT:GOSUB182                                 KL
•20 REM**READ & COMPILE MELODIES**                 OE
•22 A=B(1):FORI=1TOQM:FORJ=0TOQN-1:READN$          FC
•24 A$=LEFT$(N$,1):IFA$="0"THENL=0:GOTO44          BH
•28 N=(ASC(A$)-67)*2:N=N+(N>4)-(N<0)*13            GE
•30 B$=MID$(N$,2,1)                                GL
•32 IFB$="+"THENN=N+1                              NK
•34 IFB$="-"THENN=N-1                              PP
•36 OC=VAL(RIGHT$(N$,1))                           FF
•38 F=F(N)/(2[UPARROW](7-OC))                      IC
•40 H=INT(F/256):L=INT(F-H*256)                    DF
•42 POKEA+1,H                                      DJ
•44 POKEA,L:A=A+2:NEXT:NEXT                        DG
•46 REM**SET MELODY #0**                           HB
•48 FORI=BTOB(1)-1:POKEI,0:NEXT                    EE
•50 REM**SET SID ADDRESSES**                       KK
•52 SID=54272:FORI=0TO2:A=SID+I*7                  BN
•54 X(I)=A:Y(I)=A+1:Z(I)=A+4:NEXT                  LJ
•56 GOSUB152:POKESID+24,15                         HH
•60 REM**READ INSTRUMENTS**                        JI
•62 FORI=0TO2:READW$,A,D,S,R,PF                    DG
•64 W(I)=-(W$="T")*17-(W$="S")*33-(W$="P"
)*65-(W$="N")*129                                  OJ
•66 POKEX(I)+5,A*16+D:POKEX(I)+6,S*16+R            OH
•68 PH=INT(PF/256):POKEX(I)+3,PH:POKEX(I)
+2,PF-PH*256:NEXT                                  LA
•100 REM**PLAY TUNE**                              KP
•102 T=TI                                          KC
•104 FORI=0TO2:READM:IFM=-1THEN142                 JB
•106 PRINTM,:A(I)=B(M):NEXT                        GM
•108 FORI=0TOQN-1                                  NL
•110 T=T+P2                                        IG
•112 IFT>TITHEN112                                 HI
•114 FORJ=0TO2:L(J)=PEEK(A(J)):H(J)=PEEK(
A(J)+1)                                            NK
•116 POKEZ(J),W(J)-1:A(J)=A(J)+2:NEXT              PK
•118 T=T+P3                                        IJ
•120 IFT>TITHEN120                                 HJ
•122 FORJ=0TO2:IFL(J)=0THEN126                     GP
•124 POKEX(J),L(J):POKEY(J),H(J):POKEZ(J)
,W(J)                                              FO
•126 NEXT:NEXT:PRINTCHR$(145):GOTO104              EK
•140 REM**END**                                    CL
•142 PRINTCHR$(147):GOSUB152:END                   GI
•150 REM**SUB:CLEAR SOUND CHIP**                   PA
•152 FORI=SIDTOSID+28:POKEI,0:NEXT:RETURN          AO
•180 REM**SUB:DISPLAY TITLE**                      AJ
•182 PRINTCHR$(147)CHR$(159)                       EA
•184 PRINT"THE SIDNEY CHIPP ORCHESTRA PLA
YS[3"-"]":PRINT:PRINT                              LJ
•186 PRINTCHR$(18)Z$:PRINT:PRINT:RETURN            BE
•200 REM**FREQ DATA**                              JJ
•202 DATA34334,36376,38539,40830,43258,45
830,48556,51443,54502,57743,61176,64814            FI
•300 REM**TITLE**                                  MF
•302 DATAPACHELBEL'S CANON                         AO
•310 REM**NO.OF MELODIES,NOTES PER MELODY
**                                                 GB
•312 DATA7,32                                      FJ
•320 REM**NOTE DURATION,GATE DURATION**            FE
•322 DATA16,4                                      EO
•400 REM **MELODY DATA**                           DK
•410 REM #1                                        OO
•412 DATAC3,0,0,0,G2,0,0,0,A2,0,0,0,E2,0,
0,0                                                BP
•414 DATAF2,0,0,0,C2,0,0,0,F2,0,0,0,G2,0,
0,0                                                KO
```

```
·420 REM #2                                 ON
·422 DATAE6,0,0,0,D6,0,0,0,C6,0,0,0,B5,0,
 0,0                                        HN
·424 DATAA5,0,0,0,G5,0,0,0,A5,0,0,0,B5,0,
 0,0                                        JD
·430 REM #3                                 OM
·432 DATAC5,0,E5,0,G5,0,F5,0,E5,0,C5,0,E5
 ,0,D5,0                                    IE
·434 DATAC5,0,A4,0,C5,0,G4,0,C5,0,A4,0,C5
 ,0,D5,0                                    NG
·440 REM #4                                 OL
·442 DATAC5,B4,C5,0,G4,0,B4,0,C5,0,E5,0,G
 5,0,A5,0                                   PE
·444 DATAF5,E5,D5,F5,E5,D5,C5,E5,D5,0,A4,
 0,C5,0,B4,0                                LM
·450 REM #5                                 OK
·452 DATAC5,0,0,E5,E5,F5,E5,D5,C5,0,0,C6,
 C6,D6,C6,B5                                GE
·454 DATAA5,B5,C5,0,G5,0,C6,0,A5,0,C6,0,C
 6,0,B5,0                                   FM
·460 REM #6                                 OJ
·462 DATAC3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0   JN
·464 DATAC3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0   JN
·470 REM #7                                 OI
·472 DATAC5,0,0,0,A5,0,G5,0,G5,0,F5,0,F5,
 0,E5,0                                     BC
·474 DATAE5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0   JN
·600 REM**INSTRUMENT DATA**                 FB
·602 DATA P,0,0,15,15,4000                  MF
·604 DATA P,4,5,13,14,3700                  JH
·606 DATA P,1,6,10,11,3400                  MC
·700 REM**PLAN**                            HK
·702 DATA1,0,0                              HG
·704 DATA1,2,0                              GI
·706 DATA1,2,3                              GN
·708 DATA1,2,4                              GM
·710 DATA1,3,4                              HB
·712 DATA1,2,5                              GP
·714 DATA1,3,5                              HA
·716 DATA1,4,5                              HB
·718 DATA0,2,3                              GO
·720 DATA0,4,5                              HK
·722 DATA4,2,3                              IC
·724 DATA5,4,3                              ID
·726 DATA1,3,4                              HB
·728 DATA1,4,5                              HB
·730 DATA6,7,7                              IN
·732 DATA-1                                 OE
```



Call us the modern way—

on *Ahoy!*'s Bulletin Board System!

If your computer is equipped with a modem, you can call *Ahoy!*'s Bulletin Board System any hour of the day, any day of the week to exchange electronic mail with other Commodore users or download files like the following:

- Editorial calendar for upcoming issues
- Excerpts from future editions of *Scuttlebutt*
- Corrections to programs and articles
- Detailed descriptions of back issues
- Program disk and subscription information
- Classified advertising

Set your modem for 300 baud, full duplex, no parity, 1 stop bit, 7 word length, and dial away!

System Configuration of *Ahoy!* Bulletin Board System:

C-64 and 1650 Automodem
Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
Phone: 215-431-9100

MSD SD-2 Dual Disk Drive
MSD Systems, Inc.
10031 Monroe Street—Ste. 206
Dallas, TX 75229
Phone: 214-357-4434

Impact Printer
Fidelity Electronics, Ltd.
8800 N.W. 36th Street
Miami, FL 33178
Phone: 305-594-1000

Vortex I BBS Software
Event Horizon Corporation
Box 1327, New York, NY 10028
Phone: 212-535-0697 (voice)
212-988-4539 (data)

Sakata SC-100 Color Monitor
Sakata U.S.A. Corporation
651 Bonnie Lane
Elk Grove Village, IL 60007
Phone: 312-593-3211

CSS Monitor Stand
Computer Support Services
P.O. Box 927
Santa Barbara, CA 93102
Phone: 619-968-3040

Superstand I
A World of Plastic, Ltd.
2535 United Lane
Elk Grove Village, IL 60007
Phone: 312-860-2525

Uni-Kool C-100
Uni-Kool
909 Williamson Loop
Grants Pass, OR 97526
Phone: 503-476-1660

Electra Computer Pad
Charleswater Products, Inc.
93 Border Street
West Newton, MA 02165
Phone: 617-964-8370

Computer Stands #5933 & #0640
Royal Seating Corporation
P.O. Box 753
Cameron, TX 76520
Phone: 817-697-6421

... COMING IN THE MAY *AHOY!* (ON SALE APRIL 9) ...

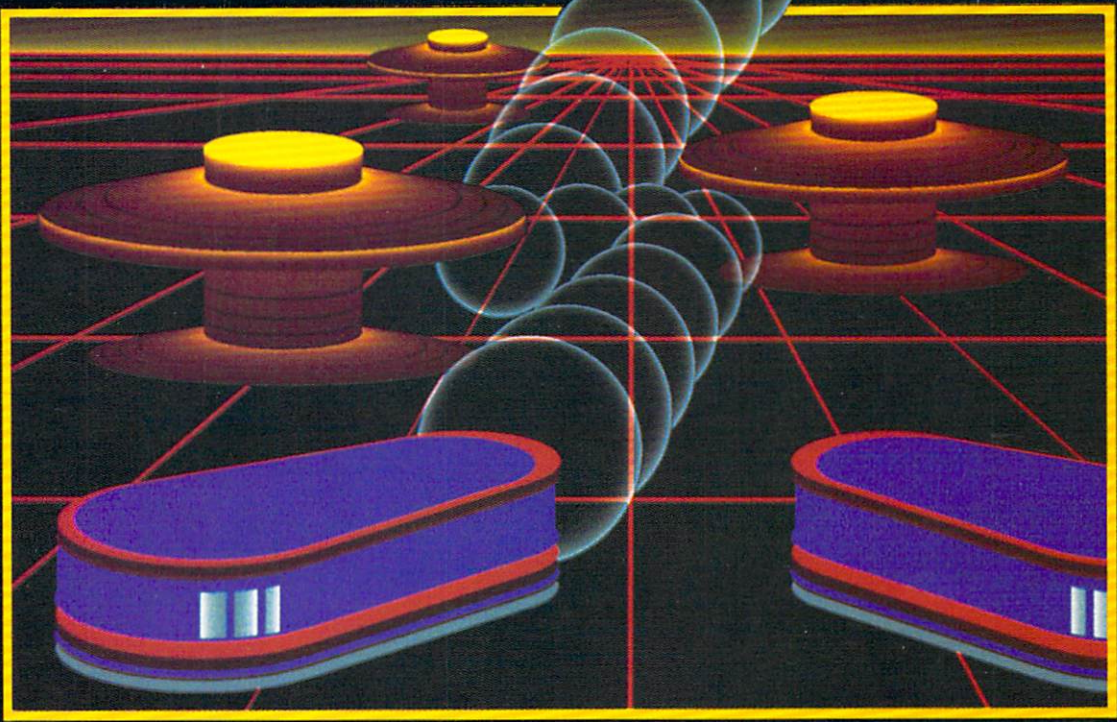
BRIGHTEN YOUR GAMES BY MOVING THROUGH COLOR MEMORY
MANIPULATING COLORS SEPARATELY FROM CHARACTERS

ASSEMBLER ESCAPADES
PUTTING ASSEMBLY LANGUAGE ROUTINES INTO BASIC

MULTIPLE METHODS OF INSTALLING A WARM BOOT SWITCH
ON YOUR VIC OR 64

For Commodore 64™ Computers

TAKE A BREAK!



WITH NIGHT MISSION

PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of **Night Mission PINBALL**, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put **Night Mission PINBALL** in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with **Night Mission PINBALL** from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.



See your dealer . . .

or write or call for more information.

Order Line: 800 / 637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Reader Service No. 124